รายงานวิจัยฉบับสมบูรณ์

โครงการย่อยที่ 1

ระบบเว็บเชิงความหมายเพื่อการจัดการและสนับสนุนการประมวลแบบขนานสำหรับการท่องเที่ยวเชิงสุขภาพ อำเภอหัวหิน

(Semantic Web Systems for Management and Parallel Processing Support for Health Tourism in Hua Hin)

โดย รศ. ดร. จันทนา จันทราพรชัย และคณะ

31 มกราคม พ.ศ. 2558

เลขที่สัญญา RDG5650033

รายงานวิจัยฉบับสมบูรณ์

โครงการย่อย 1

ระบบเว็บเชิงความหมายเพื่อการจัดการและสนับสนุนการประมวลแบบขนานสำหรับการท่องเที่ยวเชิงสุขภาพ อำเภอหัวหิน

(Semantic Web Systems for Management and Parallel Processing Support for Health Tourism in Hua Hin)

รศ. ดร. จันทนา จันทราพรชัย มหาวิทยาลัยเกษตรศาสตร์

นางสาว ชิดชนก โชคสุชาติ มหาวิทยาลัยศิลปากร

ชุดโครงการ แผนงานวิจัยระบบบริหารจัดการการท่องเที่ยวเชิงสุขภาพ

กรณีศึกษาอำเภอหัวหิน

โดยอาศัยออนโทโลยีและเทคโนโลยีการประมวลผลแบบขนานบนกลุ่มเมฆ

สนับสนุนโดยสำนักงานคณะกรรมการวิจัยแห่งชาติ (วช.) และ

สำนักงานกองทุนสนับสนุนการวิจัย (สกว.)

(ความเห็นในรายงานนี้เป็นของคณะผู้วิจัย วช.-สกว. ไม่จำเป็นต้องเห็นด้วยเสมอไป)

กิตติกรรมประกาศ

คณะผู้จัดทำโครงการวิจัย เรื่อง ระบบเว็บเชิงความหมายเพื่อการจัดการและสนับสนุนการประมวลแบบขนานสำหรับการท่องเที่ยวเชิงสุขภาพ อำเภอหัวหิน ขอขอบพระคุณบุคลากรต่างๆ ที่เกี่ยวข้องกับโครงการวิจัยนี้ โดยฉพาะผศ. ดร. ภุชงค์ อุทโยภาศ ที่ปรึกษาแผนงานวิจัยที่ให้คำแนะนำในการวิจัย และขอขอบคุณ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ และภาควิชาคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยศิลปากร ที่ให้การสนับสนุนด้านพื้นที่ และอุปกรณ์ในการดำเนินการ รวมทั้งบุคลากรเจ้าหน้าที่ธุรการที่ช่วยงานในโครงการวิจัยนี้

คณะผู้วิจัยขอขอบพระคุณสำนักงานกองทุนสนับสนุนการวิจัย (สกว.) โครงการวิจัยการท่องเที่ยวและบริการ ขอขอบพระคุณสำนักงานคณะกรรมการวิจัยแห่งชาติ (วช.) ที่ให้ทุนสนับสนุนในการการทำวิจัยครั้งนี้ ขอขอบพระคุณบริษัท กสท. โทรคมนาคมแห่งประเทศไทย จำกัด ที่ให้การสนับสนุนการใช้แพลตฟอร์ม IRIS เพื่อทดสอบระบบ

ขอขอบคุณผู้มีส่วนร่วมในงานวิจัยนี้ ได้แก่ คุณนพพร วุฒิกุล คุณมนตรี ชูภู่ และเทศบาลเมืองหัวหินที่ให้ความร่วมมือในการสนับสนุนการนำผลงานจากแผนงานวิจัยนี้ไปใช้ประโยชน์ ในการให้คำแนะนำรวมทั้งประเมินส่วนต่างๆ ของโครงการ

สุดท้ายคณะผู้วิจัยขอขอบคุณทีมงานและผู้ช่วยวิจัยทุกท่าน รวมทั้งบุพการีผู้มีพระคุณที่ให้การอุปถัมภ์ทั้งแรงกายและแรงใจแก่คณะผู้วิจัยทุกคนที่ทำให้งานวิจัยนี้สำเร็จลุล่วงไปด้วยดี

คณะผู้วิจัย

บทสรุปผู้บริหาร

ในแผนพัฒนาเศรษฐกิจและสังคมแห่งชาติ ฉบับที่ 11 (พ.ศ.2554 - พ.ศ.2559) การพัฒนาศักยภาพระดับคุณภาพบริการในอุตสาหกรรมบริการและการท่องเที่ยวโดยเฉพาะจากการท่องเที่ยวเชิงสุขภาพสามารถสร้างมูลค่าเพิ่มให้กับสาขาบริการที่เป็นมิตรกับสิ่งแวดล้อม การพำนักระยะยาวและดูแลผู้สูงอายุในท้องถิ่นด้วยความคิดสร้างสรรค์และนวัตกรรมแบบใหม่ เทคโนโลยีปัจจุบันในสาขาคอมพิวเตอร์ที่จะมีบทบาทในการปรับปรุงการให้บริการนี้ ได้แก่ การประมวลผลแบบขนานบนกลุ่มเมฆ และการบริการเว็บเชิงความหมายด้วยออนโทโลยีนั้น ซึ่งจะมีส่วนช่วยในประเด็นต่างๆ เช่น

1) ความรวดเร็วในการสืบค้นข้อมูลด้วยการประมวลผลแบบขนาน

2) คำตอบจากการสืบค้นที่ตรงตามความต้องการของผู้ค้นหาข้อมูล ด้วยการค้นหาเชิงความหมายในขอบเขตโดเมนความรู้เฉพาะทาง

3) ความสามารถในการขยายฐานข้อมูลผู้ให้บริการการท่องเที่ยวในรูปแบบต่างๆ ที่ไม่ยึดติดพื้นที่ทางกายภาพด้วยเทคโนโลยีกลุ่มเมฆ

ในโครงการวิจัยนี้ ผู้วิจัยจึงขอนำเสนอ ต้นแบบระบบบริหารจัดการการท่องเที่ยวเชิงสุขภาพโดยอาศัยการผสมผสานกับเทคโนโลยีนี้สำหรับกรณีศึกษา อำเภอหัวหิน ในโครงการย่อยนี้จะเน้นการนำเสนอข้อมูลเพื่อให้สอดคล้องกับการประมวลผลแบบขนานและประมวลผลแบบกลุ่มเมฆ พัฒนาวิธีการประมวลผลแบบขนานที่เหมาะสม มีประสิทธิภาพเพื่อการค้นหาข้อมูลที่รวดเร็ว

วัตถุประสงค์

เพื่อพัฒนาระบบการค้นหาข้อมูล สำหรับให้บริการข้อมูลการท่องเที่ยวเชิงสุขภาพสำหรับอำเภอหัวหิน โดยเน้นที่การประมวลผลข้อมูลแบบขนาน และสำหรับกลุ่มเมฆอย่างมีประสิทธิภาพ

ขอบเขตการวิจัย

พัฒนาระบบบริการข้อมูลท่องเที่ยวเชิงสุขภาพ ใช้เทคโนโลยีเว็บเชิงความหมาย และการประมวลคิวรี่แบบขนาน ทำงานได้บนกลุ่มเมฆ

ขั้นตอนการดำเนินงาน

1. สำรวจความต้องการซอฟต์แวร์ในปัจจุบัน
2. การศึกษางานวิจัยที่เกี่ยวข้องด้านการประมวลผลแบบขนาน และกลุ่มเมฆ
3. พัฒนาตัวแบบของระบบการประมวลผลแบบกลุ่มเมฆ
4. พัฒนาอัลกอริธึมการประมวลผลคิวรี่แบบขนาน และการแทนค่าข้อมูล
5. พัฒนาโปรแกรมการประมวลผลคิวรี่แบบขนาน
6. การออกแบบหน้าจอ การติดต่อผู้ใช้ ระบบออนโทโลยี
7. ออกแบบฐานข้อมูลระบบ ได้แก่ ข้อมูลผู้ใช้ระบบ ข้อมูลสถานที่ท่องเที่ยว
8. พัฒนาระบบจัดการการค้นหา พัฒนาโปรแกรม ทดลองใช้

9. เปรียบเทียบผลการค้นหาคิวรี่ ด้านประสิทธิภาพ
10. ประเมินผลการใช้งานกับผู้ใช้บทบาทต่างๆ กัน
11. เขียนสรุปรายงานและเอกสารวิชาการ
12. จัดเตรียมการอบรมผู้ใช้ จัดทำคู่มือผู้ใช้งาน

ผลการดำเนินงาน

ผลการวิจัยดำเนินงานแบ่งเป็น 3 ส่วน ดังนี้ ส่วนแรกเกี่ยวกับการสำรวจลักษณะของซอฟต์แวร์ที่ใช้ในปัจจุบันของหน่วยงาน และข้อมูลที่หน่วยงานใช้อยู่ ส่วนที่ 2 เป็นผลการประเมินประสิทธิภาพด้านเวลาของระบบเว็บเชิงความหมาย การประมวลผลแบบขนาน ส่วนที่ 3 เป็นผลการประเมินประสิทธิภาพด้วยเวลาสำหรับการประมวลผลการดึงข้อมูล และค้นหาข้อมูลแบบขนาน ส่วนการประเมินผู้ใช้งานนั้นอยู่ในภาคผนวกของรายงานแผนงานวิจัย จึงไม่ขอกล่าวถึงในที่นี้

1. การสำรวจข้อมูลของเทศบาลด้านลักษณะหน่วยงาน ข้อมูลที่มีและซอฟต์แวร์ต่างๆ

คณะผู้วิจัยได้ สอบถามข้อมูลซอฟต์แวร์ที่ใช้ภายในหน่วยงานเทศบาล และการทำงานของหน่วยงาน แหล่งข้อมูลต่างๆ ของการท่องเที่ยวที่ทางเทศบาลใช้อยู่ ด้วยการสัมภาษณ์และแบบสอบถาม โดยรายละเอียดอยู่ในรายงานแผนการวิจัย

ข้อมูลเกี่ยวข้องกับหน่วยงานเทศบาลเมืองหัวหิน

ผู้ที่เกี่ยวข้องกับโครงงานนี้ในหน่วยงานของเทศบาลเมือง คือ นายกเทศมนตรีมอบหมายงานให้รองนายกเทศมนตรีด้านการท่องเที่ยว คือคุณมนตรี ชูภู่ เป็นผู้อนุมัติการประเมินระบบนี้ รองนายกเทศมนตรีผู้รับผิดชอบด้านการท่องเที่ยว ได้มอบหมายฝ่ายอำนวยการ คือคุณเบญจวรรณ เมธาวิกุล มีหน้าที่ควบคุมดูแลและรับผิดชอบการปฏิบัติงานในหน้าที่ที่เกี่ยวข้องกับงานส่งเสริมการท่องเที่ยว มีหัวหน้างานคือ คุณอุกฤษฎ์ ป้อทองคำ มีพนักงาน 7 คน สำหรับงานส่งเสริมการท่องเที่ยว มีหน้าที่เกี่ยวกับ

- งานส่งเสริมและเผยแพร่การท่องเที่ยวของเทศบาลภายในประเทศ
- งานควบคุมกิจการบริการท่องเที่ยวภายในประเทศของเทศบาลและหน่วยงานอื่น
- งานต้อนรับ แนะนำ อำนวยความสะดวกและบริการนำเที่ยว รวมทั้งดำเนินธุรกิจการท่องเที่ยว
- งานประชาสัมพันธ์ ให้คำแนะนำเผยแพร่ความรู้เกี่ยวกับแหล่งท่องเที่ยวหรือกิจการ ที่เกี่ยวกับการท่องเที่ยวภายในจังหวัดและเทศบาล
- งานจัดทำรายงานและสถิติการเคลื่อนไหวเกี่ยวกับธุรกิจการท่องเที่ยว
- งานอื่นที่เกี่ยวข้องหรือตามที่ได้รับมอบหมาย

ข้อมูลที่ได้จากการสำรวจความต้องการและข้อมูลที่มีอยู่

รายละเอียดของการสำรวจความต้องการซอฟต์แวร์จากหน่วยงานเทศบาลเมืองหัวหินสำหรับการสำรวจข้อมูลในพื้นที่นั้นได้ทำ 2 ครั้ง นั่นคือ ครั้งที่ 1 เมื่อวันที่ 23 สิงหาคม พ.ศ. 2556 และได้กำหนดวันนัดหมายเพื่อพูดคุยรายละเอียดกันในเดือนกันยายน แต่ได้มีการเลื่อนเป็นครั้งที่ 2 วันที่ 4 ตุลาคม พ.ศ. 2556 ดังรายละเอียดดังต่อไปนี้

ระดับผู้บริหารนั้นมีความต้องการใช้ข้อมูลที่ผ่านการรวบรวมอย่างเป็นระบบ เนื่องจากการนำเสนอข้อมูลจากหน่วยงานต่อสาธารณชนนั้นควรมีรายละเอียดรอบด้านมากขึ้น
การสัมภาษณ์ผู้อำนวยการฝ่ายอำนวยการ และสำรวจจากเทศบาลเมืองหัวหิน หัวหน้างานส่งเสริมการท่องเที่ยวและเจ้าหน้าที่การท่องเที่ยวแห่งประเทศไทยในเขตเมืองหัวหินและพนักงานปฏิบัติการ คุณ เกษมศักดิ์ ปัดออด มีดังแบบสำรวจความต้องการการใช้งานซอฟต์แวร์ของระบบการจัดการท่องเที่ยวเชิงสุขภาพ สรุปผลการสำรวจได้ดังนี้

- ระดับหน่วยงานส่งเสริมการท่องเที่ยวมีศูนย์บริการนักท่องเที่ยว 2 แห่งคือเทศบาลเมืองหัวหินและหอนาฬิกา
- การรวบรวมข้อมูลเป็นการทำงานแบบระบบมือ
- การค้นหาข้อมูลเกี่ยวกับที่พัก ร้านอาหาร สถานที่แจ้งภัย ส่วนมากใช้ระบบเดินสำรวจหรือค้นหาทางเว็บไซต์เฉพาะที่ ในขณะที่สถานพยาบาล สปา นวดแผนไทย อายุรเวช นั้นไม่เคยสำรวจ
- การติดต่อสื่อสารระหว่างหน่วยงานเทศบาลและที่พักชั่วคราวอาทิโรงแรมในพื้นที่มักใช้โทรศัพท์และการเดินเอกสาร สำหรับการใช้งานโปรแกรมจองโรงแรมหรือเว็บไซต์เกี่ยวกับการท่องเที่ยว เช่น Agoda, TripAdvisor, AtSiam นั้น ไม่เคยใช้ สำหรับโปรแกรมการค้นหาเว็บไซต์นั้นใช้กูเกิล ส่วนเครือข่ายสังคมออนไลน์นั้นใช้เฟสบุ๊คเพื่อกระจายข่าวสารเทศบาลและการท่องเที่ยว
- สำหรับเครื่องมือที่ใช้ค้นหาผ่านอินเทอร์เน็ตคือคอมพิวเตอร์ส่วนบุคคลที่เชื่อมต่ออินเตอร์เน็ตทั้งแลนและระบบไร้สาย
- การทำรายงานนำเสนอผู้บริหารประกอบด้วยเชื้อชาติของนักท่องเที่ยวและจำนวนผู้สอบถามเป็นรายวันรายเดือนและรายปีผ่านโปรแกรมไมโครซอฟต์เอ็กเซล
- สำหรับเอกสารเผยแพร่ข้อมูลจัดทำเป็นแผ่นพับสองภาษาคือภาษาไทยและภาษาอังกฤษและมีการปรับปรุงประมาณปีละครั้ง
- สิ่งที่ต้องการจากระบบบริหารจัดการการท่องเที่ยวเชิงสุขภาพคือข้อมูลโรงแรมเบื้องต้นเพิ่มเติมจากการการลงพื้นที่เองและจากกูเกิล ประโยชน์ที่จะนำไปใช้นอกจากรายงานสำหรับผู้บริหารคือ เป็นแหล่งข้อมูลสำหรับนักวิจัย นักเรียน นักศึกษา ในกรณีที่มีการร้องขอข้อมูลอย่างละเอียด
- ข้อมูลที่ร้องขอเพิ่มเติมคือหมายเลขโทรศัพท์และอีเมล์ของสถานทูต ข้อมูลสำนักงานตรวจคนเข้าเมืองเพื่อต่อวีซ่า
- ต้องการใช้ระบบบริหารจัดการการท่องเที่ยวเชิงสุขภาพ เพื่อรวบรวมข้อมูลร้อยละ 45เพื่อตรวจสอบข้อมูลร้อยละ 45 ใช้ประกอบรายงานสำหรับผู้บริหารร้อยละ 10
- หากโปรแกรมเสร็จแล้วมีความต้องการอบรมเพื่อใช้งานโปรแกรม

นอกจากนี้ได้มีการสอบถามข้อมูลการวางแผนท่องเที่ยวจากสำนักงานการท่องเที่ยวแห่งประเทศไทย สำนักงานประจวบคีรีขันธ์วันที่ 23 สิงหาคม พ.ศ. 2556 ในเรื่องการวางแผนการท่องเที่ยวสำหรับ

นักท่องเที่ยวทั้งชาวไทยและชาวต่างประเทศซึ่งมีการเก็บข้อมูลนักท่องเที่ยวที่คล้ายคลึงกับเทศบาลเมืองคือ เชื้อชาติและจำนวนนักท่องเที่ยวที่มาสอบถามรายวัน สำหรับข้อมูลรายได้และนักท่องเที่ยวโดยรวมของภาคกลางจะได้รับมาจากการท่องเที่ยวแห่งประเทศไทยสำนักงานใหญ่การวางแผนท่องเที่ยวในส่วนข้อมูลทั่วไปและกิจกรรมการท่องเที่ยวที่เผยแพร่นั้นเจ้าหน้าที่เป็นผู้รวบรวมขึ้นเองส่วนเครื่องมือค้นหาเว็บไซต์นั้นใช้งานผ่านกูเกิลสำหรับรายละเอียดของหน้าที่สำนักงาน กิจกรรมการท่องเที่ยว รายละเอียดนักท่องเที่ยวและแผนการท่องเที่ยวมีให้ตามแผ่นพับของ ททท.

ข้อมูลจากหน่วยงานเพื่อตรวจสอบความถูกต้อง

สำหรับสถานบริการสปาหรือนวดแผนโบราณที่เป็นเป้าหมายของงานวิจัย ในกรณีที่เป็นโรงแรมหรือที่พักขนาดใหญ่ที่มีอยู่ใน Agoda หรือ TripAdvisor นั้น จะถูกดึงเข้าสู่ระบบโดยโปรแกรมคอมพิวเตอร์ อย่างไรก็ตามยังมีสถานบริการริมชายหาดและสถานบริการเชิงสุขภาพขนาดเล็ก ซึ่งผู้อำนวยการฝ่ายอำนวยการได้กรุณาให้ความรู้ในไว้ดังนี้

- กรณีที่เป็นการนวดริมชายหาดของเอกชนนั้นสามารถตรวจสอบได้กับเทศกิจ
- กรณีที่เป็นร้านขนาดเล็ก จะมีการขึ้นทะเบียนกับกองสาธารณสุข สิ่งที่ได้คือชื่อร้านและที่อยู่
- สำหรับพิกัดที่ตั้งนั้นงานแผนที่ภาษีมีบันทึกไว้ในระบบพิกัดกริดแบบ UTM รายชื่อผู้เสียภาษีและที่อยู่
- เมื่อต้องการนำมาตรวจสอบกับระบบที่ผู้วิจัยเก็บไว้เพื่อให้ได้ชื่อร้านและพิกัดละติจูด/ลองจิจูดต้องนำข้อมูลมาจากทั้ง 3 หน่วยงานนี้มาตรวจสอบร่วมกัน

2. ผลการประเมินประสิทธิภาพ

การวัดประสิทธิภาพด้านเวลามี 2 แบบ แบบแรกเป็นการวัดประสิทธิภาพด้านเวลาการค้นหาด้วยเว็บเชิงความหมายโดยค้นหาในออนโทโลยี และแบบที่สองเป็นการค้นหาสายอักขระด้วยการประมวลผลแบบขนานด้วยหน่วยประมวลผลกราฟิก

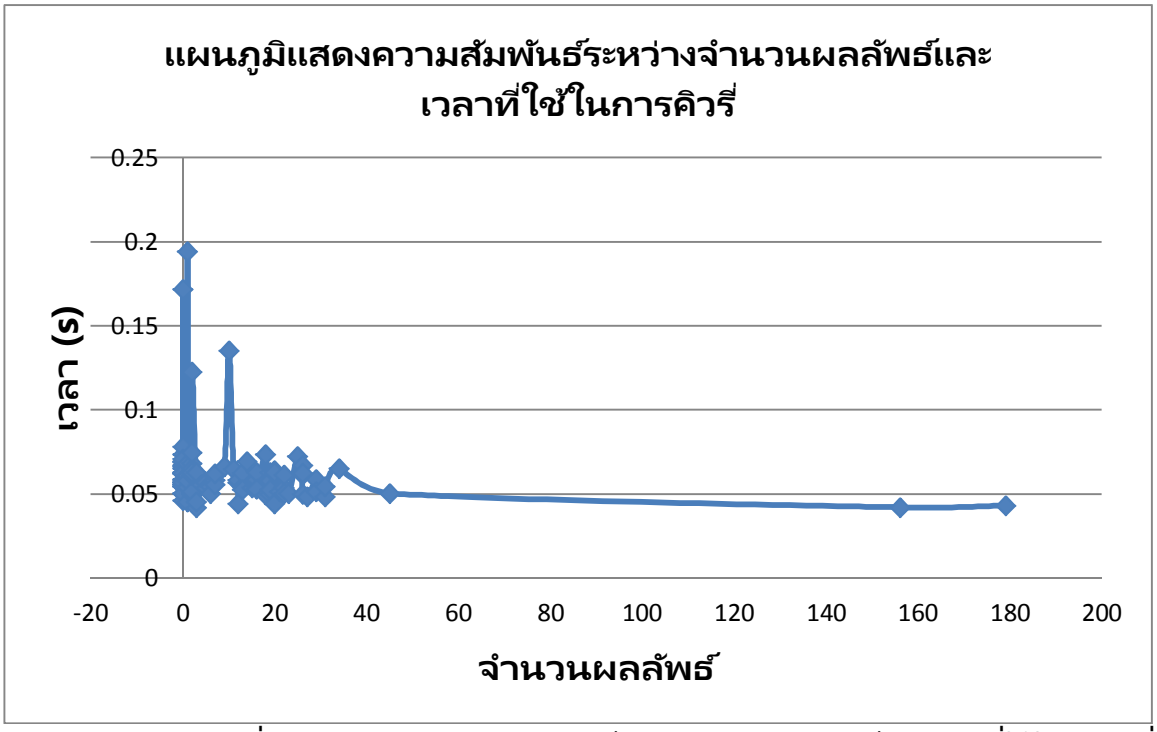
เวลาการค้นหาด้วยเว็บเชิงความหมาย

ในการทดสอบประสิทธิภาพ คณะผู้วิจัยได้ทดสอบวัดความเร็วคิวรี่จากจำนวนคิวรี่ 100 ตัวอย่าง ทั้งที่ไม่มีความซับซ้อน มีความซับซ้อนน้อยไปจนถึงตัวอย่างที่มีความซับซ้อนมากเพื่อวัดเวลาที่ใช้ในการคิวรี่ข้อมูล ตารางที่ 1 แสดงรายละเอียดข้อมูลออนโทโลยี

ตารางที่ 1 ข้อมูลออนโทโลยี

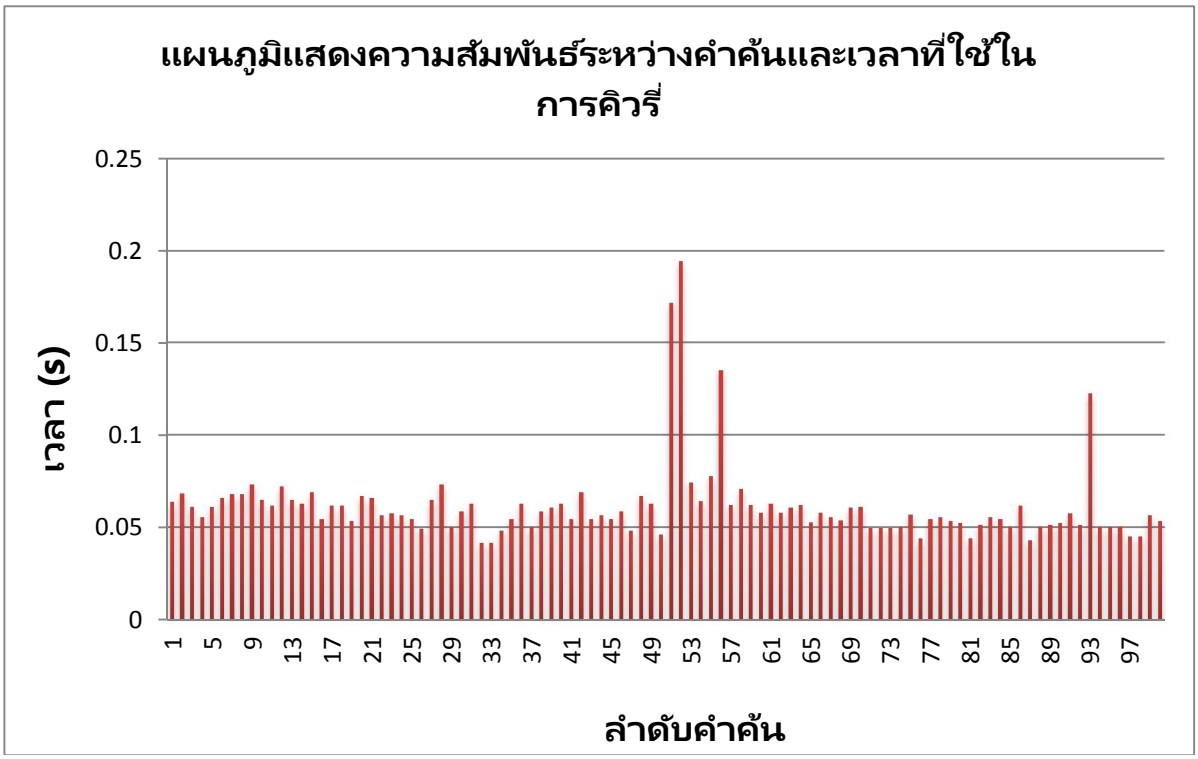
ข้อมูลทั้งหมดในออนโทโลยี	**จำนวน**
Axiom (ข้อกำหนดความสัมพันธ์)	24663
Logical axiom (ความสัมพันธ์เชิงตรรกะ)	21094
Class (คลาส)	229
Object Property (ความสัมพันธ์)	33
Data Property (คุณสมบัติ)	29
Individual (สมาชิกของคลาส)	3008
ข้อจำกัดความสัมพันธ์ในคลาส	**จำนวน**
SubClassOf (คลาสย่อย)	227
EquivalentClasses (คลาสเทียบเท่า)	19
DisjointClasses (คลาสที่ไม่เกี่ยวข้องกัน)	236
Hidden GCI (General Concept Inclusions)	13
ข้อจำกัดความสัมพันธ์ใน Object Property	**จำนวน**
SubObjectPropertyOf (ความสัมพันธ์ย่อย)	2
EquivalentObjectProperties (ความสัมพันธ์เทียบเท่า)	2
InverseObjectProperties (ความสัมพันธ์ที่แบบผันกลับ)	16
FunctionalObjectProperty (ความสัมพันธ์ระหว่างกันแบบหนึ่งต่อหนึ่ง)	2
TransitiveObjectProperty (ความสัมพันธ์ที่มีการส่งต่อไปเรื่อยๆ)	4
ObjectPropertyDomain (สมาชิกภายในเซตข้อมูลที่ใช้จับคู่กับอีกเซตหนึ่ง	33
ObjectPropertyRange (เซตข้อมูลที่ถูกจับคู่โดยความสัมพันธ์)	26
ข้อจำกัดความสัมพันธ์ใน Annotation	**จำนวน**
AnnotationAssertion (คำอธิบายประกอบคุณสมบัติ ความสัมพันธ์ และข้อมูล	271
ข้อจำกัดความสัมพันธ์ใน Individual	
ClassAssertion (การกำหนดสมาชิกในคลาส	3076
ObjectPropertyAssertion (การกำหนดความสัมพันธ์ของข้อมูล)	9943
DataPropertyAssertion (การระบุคุณสมบัติของข้อมูล)	7454

ผลการทดลองวัดเวลาที่ใช้ในการค้นหาพบว่า แต่ละตัวอย่างที่ใช้ทดลองคิวรี่ใช้เวลาในการค้นหาใกล้เคียงกัน และใช้เวลาไม่มาก โดยเฉลี่ยใช้เวลาน้อยกว่า 1 วินาที (เวลาโดยเฉลี่ย 0.061618 วินาที)ทั้งนี้ความซับซ้อนของคำค้นและจำนวนผลลัพธ์ของการสืบค้นไม่ส่งผลให้ใช้เวลาในการค้นหาแตกต่างกันมาก



ภาพที่ 1 แผนภูมิแสดงความสัมพันธ์ระหว่างจำนวนผลลัพธ์และเวลาที่ใช้ในการคิวรี่

ภาพที่ 1 แสดงแผนภูมิความสัมพันธ์ระหว่างจำนวนผลลัพธ์และเวลาที่ใช้ในการคิวรี่ แสดงให้เห็นว่าจำนวนผลลัพธ์ของการสืบค้นไม่ส่งผลให้เวลาที่ใช้ในการคิวรี่แตกต่างกันมาก แต่ละตัวอย่างที่ใช้ทดลองในการคิวรี่มีเวลาใกล้เคียงกัน และใช้เวลาโดยเฉลี่ยไม่ถึง 1 วินาที(เวลาเฉลี่ย 0.061618 วินาที) ทั้งนี้จากแผนภูมิจะเห็นได้ว่ากราฟที่ได้มีลักษณะเป็นเส้นตรง และมีบางข้อมูลที่มีค่าเวลาที่ใช้ในการคิวรี่มากกว่าข้อมูลอื่นๆ เนื่องจากมีปัจจัยภายนอกที่ส่งผลให้เวลาที่ใช้ในการคิวรี่มากขึ้น เช่น ความเร็วของอินเทอร์เน็ต เป็นต้น



ภาพที่ 2 แผนภูมิแสดงความสัมพันธ์ระหว่างคำค้นและเวลาที่ใช้ในการคิวรี่

ภาพที่ 2 แผนภูมิแสดงความสัมพันธ์ระหว่างคำค้นและเวลาที่ใช้ในการคิวรี่ แสดงให้เห็นว่าความซับซ้อนของคำค้นไม่ส่งผลให้เวลาที่ใช้ในการคิวรี่แตกต่างกันมาก แต่ละตัวอย่างที่ใช้ทดลองในการคิวรี่มีเวลาใกล้เคียงกันและใช้เวลาโดยเฉลี่ยไม่ถึง 1 วินาที (เวลาเฉลี่ย 0.061618 วินาที) ทั้งนี้จากแผนภูมิจะเห็นได้ว่าความสูงของกราฟที่ได้มีลักษณะเป็นเส้นตรง และมีบางข้อมูลที่มีค่าเวลาที่ใช้ในการคิวรี่มากกว่าข้อมูลอื่นๆ เนื่องจากมีปัจจัยภายนอกที่ส่งผลให้เวลาที่ใช้ในการคิวรี่มากขึ้น เช่น ความเร็วของอินเทอร์เน็ต เป็นต้น

การประเมินเวลาของการนำเข้าข้อมูล

ส่วนเวลาการค้นหาด้วย Java Concurrency โดยการแตกเธรดพบว่าเมื่อมีจำนวนเธรดมากขึ้น จะช่วยให้เวลาที่ใช้ในการคิวรี่ลดลง โดยในเบื้องต้น ทดสอบกับข้อมูลจาก DBpedia BTC2012[1] โดยแฟ้มข้อมูลมีขนาด 4GB จำนวน 21,398,608 Triples

คณะผู้วิจัยทดสอบเทียบเวลาการทำงานของการค้นหาเมื่อใช้เครื่องคอมพิวเตอร์เพียง 1 โหนด กับเวลาเมื่อใช้เครื่องคอมพิวเตอร์จำนวน 4 โหนด โดยงาน Map มีจำนวนตั้งแต่ 4 ถึง 61 และขนาดบล๊อกข้อมูลมีขนาดตั้งแต่ 64, 128, 256, 512 และ 1024 MB และทดสอบกับงาน Reduce 1 หรือ 2 งาน ผลการทดลองพบว่าเวลาที่ใช้ลดลงเมื่อจำนวนงาน Map ลดลง และเมื่อขนาดบล๊อกใหญ่ขึ้น สำหรับคลัสเตอร์ขนาดเล็กนี้ จากงานวิจัยในอดีตพบว่าจำนวนงาน Map จะต้องสอดคล้องกับจำนวนคอร์ทั้งหมดที่มี และจำนวนงาน Reduce มีเพียง 1 หรือ 2 งาน ก็เพียงพอ แต่ควรกำหนดขนาดบล๊อก ให้ใหญ่ให้เหมาะสมเพียงพอเพื่อให้เวลาการทำ I/O ไม่มากเกินกว่าเวลาคำนวณในบล๊อกนั้นจริงๆ

ผลการทดสอบพบว่าประสิทธิภาพด้านเวลาของการทำงานบน 1 โหนด กับการทำงานบน 4 โหนด พบว่าการทำงานบน 4 โหนดทำงานได้เร็วกว่า เมื่อให้จำนวนงาน Map เป็น 8 ถ้ามีจำนวนโหนดมากขึ้น ก็จะเห็นผลด้านเวลาดีขึ้น และเมื่อกำหนดขนาดของแฟ้มให้ใหญ่ในระดับ GB การทำงานของ MapReduce เป็นการทำงานที่เน้นข้อมูลมากๆ และมีการ Overlap การคำนวณระหว่างงานเมื่องาน Map ทำงานเสร็จ Reduce ก็ทำงานต่อได้ และงาน Map อื่นๆ ก็ทำงานต่อไป จึงจะทำให้การทำงานนั้นต่อเนื่องงาน Map จึงต้องประมวลผลกับข้อมูลจำนวนมาก นอกจากนี้เวลาในการเริ่มต้นต่างๆ เช่นการเริ่มต้นของ Java Virtual Machine (JVM) ก็ไม่ต่ำกว่า 0.3-0.5 วินาที[2,3] ดังนั้น ขนาดข้อมูลในบล๊อกควรใหญ่พอเพียงด้วยให้งาน Map ทั้งหลายทำงานพร้อมกันด้วย แต่ก็ไม่ทำให้เกิดจำนวนงาน Map มากเกินไปเพราะจะให้ Overhead มากเนื่องจากมีงานในระบบเป็นจำนวนงานเกินไปเทียบกับจำนวนแกนจริงของเครื่องคอมพิวเตอร์

เมื่อเทียบกับเวลาการรวมแฟ้มข้อมูลทั่วไปด้วยโปรแกรม Java แบบลำดับจะใช้เวลามากกว่า 10 นาที สำหรับข้อมูลขนาด 4GB หรือบางครั้งอาจจะรันไม่ได้เลย การปรับพารามิเตอร์ต่างๆ เช่นจำนวนงาน Map จำนวนงาน Reduce ขนาดของบล๊อก ขนาดของแฟ้มข้อมูลทั้งหมด จำนวน VM และอื่นๆ ของ Hadoop ให้เหมาะกับคลัสเตอร์ที่มีค่อยข้างขึ้นกับลักษณะของงานและคลัสเตอร์อย่างมาก

การประเมินเวลาของการค้นหาด้วยการแทนค่าด้วยต้นไม้ไบนารี่

เมื่อใช้การแทนค่าแบบไบนารี่หรือต้นไม้ 0-1 จะเห็นว่าจะมีขั้นตอนเพิ่มกว่าการใช้รูปแบบ RDF ทั่วไป เพราะจำต้องมีการแปลงจาก RDF เป็น HDT และจาก HDT เป็นไบนารี่ จากนั้นจึงนำข้อมูลที่ผ่านการแปลงเรียบร้อยแล้วเข้าหน่วยประมวลผลกราฟิกส์ เพื่อทำการค้นหา การแปลงเป็นไบนารีย่อมได้เปรียบในการย่อขนาดข้อมูลซึ่งจะทำให้สามารถนำข้อมูลเข้าไปในหน่วยประมวลผลกราฟิกส์ได้มากกว่ารูปแบบ RDF

---

[1] http://km.aifb.kit.edu/projects/btc-2012/

[2] http://stackoverflow.com/questions/10862096/hadoop-map-reduce-how-to-speed-up-job-launch-setup.

[3] http://blog.griddynamics.com/2010/07/war-story-optimizing-one-hadoop-job.html.

คณะผู้วิจัยได้ทำการเปรียบเทียบจำนวนเทอมในหน่วยหลักล้านของแฟ้มข้อมูลในรูปแบบ RDF รูปแบบ HDT และรูปแบบไบนารี่ที่นำเสนอ จะเห็นว่าในงานวิจัยนี้นำเสนอรูปแบบข้อมูลที่มีขนาดเล็กกว่าแบบดั้งเดิมและแบบ HDT ในทุกกรณี

จากนั้นทำการทดสอบวัดเวลาในการโอนย้ายข้อมูลจากหน่วยประมวลผลกลางไปหน่วยประมวลผลกราฟิกส์ เมื่อข้อมูลเป็นแบบไบนารีจะทำให้สามารถโอนย้ายข้อมูลได้เร็วกว่า

จากการวัดเวลาที่ใช้ในการค้นหาข้อมูลในหน่วยประมวลผลกลางและหน่วยประมวลผลกราฟิกส์ตามลำดับ เมื่อข้อมูลอยู่ในรูปแบบ HDT ที่ได้กำหนดการสุ่ม ID ของ Subject, Predicate, Object จำนวน 1,024 ตัว และวัดเวลาเฉลี่ยของการค้นหาจำนวน 25 ครั้ง เมื่อข้อมูลอยู่ในรูปแบบ SPO โดยใช้ข้อมูลต่อไปนี้ในการทดสอบ swdf, wn3.0, Freebase และ DBPedia จะเห็นว่าวิธีการแบบไบนารี สามารถนำเข้าข้อมูลได้มากกว่า และค้นหาข้อมูลได้พร้อมกันหลายตำแหน่ง ทำให้ใช้เวลาน้อยกว่า อย่างไรก็ดีการใช้วิธีนี้จำเป็นต้องมีเวลาที่ต้องใช้ในการแปลงข้อมูลมาพิจารณาด้วย

3. การประเมินความพึงพอใจของผู้ใช้บริการเว็บเชิงความหมาย

ในการวิจัยได้ทำการประเมินความพึงพอใจของผู้ใช้งานระบบเว็บเชิงความหมายกำหนดให้ผู้ใช้งานระบบจำนวน 100 คนทำแบบสอบถามเพื่อประเมินประสิทธิภาพการทำงานของระบบใน 3 ด้านคือ

- ความพึงพอใจด้านคุณภาพของข้อมูล
- ความพึงพอใจด้านการออกแบบและการจัดรูปแบบ
- ความพึงพอใจด้านประโยชน์ของข้อมูลที่ได้รับ

ผลการประเมินความพึงพอใจด้านคุณภาพของข้อมูลจะเห็นได้ว่า ค่าเฉลี่ยของความพึงพอใจด้านคุณภาพของข้อมูลเท่ากับ 3.96 และมีค่าเฉลี่ยของส่วนเบี่ยงเบนมาตรฐานเท่ากับ 0.57 ดังนั้นระบบที่ได้พัฒนาขึ้นมีประสิทธิภาพในด้านคุณภาพของข้อมูลอยู่ในระดับดี

ผลการประเมินความพึงพอใจด้านการออกแบบและการจัดรูปแบบ การประเมินความพึงพอใจด้านการออกแบบและการจัดรูปแบบซึ่งได้ผลลัพธ์ว่าความพึงพอใจด้านนี้มีค่าเฉลี่ยเท่ากับ 4.08 และมีค่าส่วนเบี่ยงเบนมาตรฐานโดยเฉลี่ยเท่ากับ 0.54 ดังนั้นระบบที่ได้พัฒนาขึ้นมีประสิทธิภาพในด้านการออกแบบและการจัดรูปแบบอยู่ในระดับดี

ผลการประเมินความพึงพอใจด้านประโยชน์ของข้อมูลที่ได้รับ การประเมินความพึงพอใจด้านประโยชน์ของข้อมูลที่ได้รับซึ่งได้ผลลัพธ์ว่าความพึงพอใจด้านนี้มีค่าเฉลี่ยเท่ากับ 3.90 และมีค่าส่วนเบี่ยงเบนมาตรฐานโดยเฉลี่ยเท่ากับ 0.54 ดังนั้นระบบที่ได้พัฒนาขึ้นมีประสิทธิภาพในด้านประโยชน์ของข้อมูลที่ได้รับอยู่ในระดับดี

ผลการประเมินความพึงพอใจโดยรวม ซึ่งได้ผลลัพธ์ว่าความพึงพอใจโดยรวมมีค่าเฉลี่ยเท่ากับ 3.96 และมีค่าส่วนเบี่ยงเบนมาตรฐานโดยเฉลี่ยเท่ากับ 0.53 ดังนั้นระบบที่ได้พัฒนาขึ้นมีประสิทธิภาพโดยภาพรวมอยู่ในระดับดี

บทสรุปและข้อเสนอแนะ

ในโครงการย่อยนี้ ได้ทำการสำรวจความต้องการของซอฟต์แวร์ปัจจุบัน รวมทั้งลักษณะโครงสร้างการทำงานของหน่วยงานเทศบาลเมือง พบว่าการค้นหาข้อมูลส่วนใหญ่เป็นการค้นหาจากเว็บไซต์ทั่วไปเช่น Agoda, TripAdvisor ผ่าน Google Search Engine เป็นต้น นอกจากนี้ข้อมูลอื่นๆ มาจากการเดินสำรวจ ซอฟต์แวร์ในการจัดเก็บข้อมูล และเก็บสถิติได้แก่ Excel และ Word เพื่อทำรายงานสรุปต่างๆ แหล่งข้อมูลต่างๆ ในการ

รวบรวมข้อมูลสถานประกอบการได้แก่ จากเว็บไซต์ Agoda, TripAdvisor, AtSiam, HeyHuaHin และจาก Official Website ของสถานบริการ รวมทั้งการเดินสำรวจในพื้นที่หัวหินและชะอำ ข้อมูลเหล่านี้ถูกนำเข้าออนโทโลยี เชื่อมต่อกับหน้าจอผู้ใช้งานที่ได้ออกแบบไว้ เพื่อสามารถค้นหาได้ สำหรับการประมวลผลการค้นหา เมื่อผู้ใช้ระบุเงื่อนไขการค้นหาหาจะถูกแปลงเป็นคิวรี่ส่งไปยังมิดเดิลแวร์เพื่อติดต่อกับออนโทโลยี และค้นหาข้อมูลที่ต้องการขึ้นมาแสดงผล

คณะผู้วิจัยการนำข้อมูลเข้าแบบขนานด้วยเทคโนโลยี Java แบบพร้อมกัน (Concurrent) และ MapReduce โดยพบว่าการนำเข้าข้อมูลโดยการค้นหาเว็บแบบพร้อมกัน ทำให้ทำงานได้เร็วขึ้น ประมาณ 3-4 เท่าสำหรับกรณีใช้ 4 เธรด และการทำงานแบบ MapReduce จะช่วยทำการรวมข้อมูลจากแฟ้มข้อมูลขนาดใหญ่ เช่น 4 GB ได้ดีภายในเวลาไม่เกิน 7 วินาที เทียบกับวิธีปกติโดยโปรแกรมภาษา Java แบบ 1 เธรดที่ใช้เวลาร่วมสิบกว่านาที นอกจากนี้คณะผู้วิจัยได้ทำการวัดประสิทธิภาพการค้นหาจากออนโทโลยี โดยจะพบว่าเวลาเฉลี่ยในการค้นหาไม่เกิน 30 วินาที และคณะผู้วิจัยได้ทำการนำเสนอโมเดลการค้นหาด้วยวิธีการแบบขนานในโมเดลต้นไม้แบบไบนารีกับหน่วยประมวลผลกราฟิกส์เพื่อประมวลผลแบบขนาน โดยนำข้อมูลออนโทโลยีในรูป Triple เข้ามาในหน่วยประมวลผลกราฟิกส์เพื่อค้นหาแบบพร้อมกัน ซึ่งสามารถนำเข้าข้อมูลได้มากขึ้นและนำมาค้นหาได้เร็วขึ้นนั่นเอง

คณะผู้วิจัยได้ประเมินความพึงพอใจจากผู้ใช้จำนวน 200 คนเพื่อวัดความพึงพอใจของข้อมูลและการออกแบบหน้าเว็บอีกด้วย การออกแบบระบบด้วยออนโทโลยีให้ข้อดี ในการค้นหาข้อมูลต่างๆ ได้ตรงตามความต้องการของผู้ใช้มากขึ้น เนื่องจากเป็นการค้นหาตามความหมายของคำค้นในบริบทที่ผู้ใช้ต้องการจริง อย่างไรก็ดีเครื่องมือที่สนับสนุนการออกแบบออนโทโลยียังมีขีดจำกัดในด้านต่างๆ เช่น กลไกการอนุมาน ลอจิกที่รองรับความเสถียรของระบบเชื่อมต่อ และการทำงานที่ช้ากว่าระบบฐานข้อมูลเชิงสัมพันธ์ที่มีอยู่ ดังนั้นการพัฒนาอย่างต่อเนื่องของเทคโนโลยีเหล่านี้อาจจะทำให้ระบบค้นหาเชิงความหมายสมบูรณ์ขึ้น

นอกจากนี้การรองรับการประมวลผลแบบขนานของเครื่องมือซอฟต์แวร์ที่มีอยู่ยังมีขีดจำกัดทำให้เมื่อต้องการใช้กลไกการค้นหาแบบขนานมาช่วย ต้องทำการออกแบบการเชื่อมต่อ รวมทั้งเวลาการประมวลผลขั้นก่อนการค้นหาหรือ Preprocess ต่างๆ ค่อนข้างมากซึ่งเป็น Overhead ในการทำงานโดยรวมอยู่

ในงานวิจัยในอนาคตการพัฒนาโครงสร้างข้อมูลการแทนค่าออนโทโลยีพื้นฐานนี้ให้เหมาะกับการประมวลผลแบบขนานไม่ว่าจะด้วยลักษณะคลัสเตอร์ หรือหน่วยประมวลผลกราฟิกก็ตาม ไลบารีต่างๆ ที่รองรับจะมีส่วนสำคัญในการเชื่อมต่อการประมวลผลออนโทโลยีกับเว็บเชิงความหมายให้เป็นอัตโนมัติและด้วยประสิทธิภาพที่ดีต่อไป

บทคัดย่อ

ในงานวิจัยนี้ คณะผู้วิจัยได้พัฒนาระบบค้นหาสำหรับการบริการข้อมูลเชิงสุขภาพในอำเภอหัวหิน ระบบค้นหาได้อาศัยเทคโนโลยีการประมวลแบบขนานและกลุ่มเมฆ เชื่อมต่อกับข้อมูลท่องเที่ยวเชิงสุขภาพที่จัดเก็บในรูปแบบออนโทโลยี สนับสนุนการเทคโนโลยีเชิงความหมาย คณะผู้วิจัยได้นำระบบขึ้นไปวางบนกลุ่มเมฆในลักษณะงานประยุกต์โดยการสนับสนุนของ กสท. (ตะวันตก) ภายในแพลตฟอร์มของ IRIS

การประมวลแบบขนานที่ใช้ในงานวิจัยนี้มี 2 ด้านได้แก่ ด้านเสาะแสวงหา ดึงข้อมูล และเก็บข้อมูล โดยใช้เทคโนโลยี Java Concurrent (Java 1.7) ร่วมกับการพัฒนาโปรแกรมสำหรับการจัดทำกลไกการดึงรวบรวมข้อมูลหน้าเว็บ นอกจากนี้ได้ใช้เฟรมเวิร์ก MapReduce ในการพัฒนาส่วนของการรวมข้อมูล อีกด้านได้แก่การออกแบบการแทนค่าข้อมูลสำหรับการประมวลคิวรี่ภายในสถาปัตยกรรม GPU ซึ่งเป็นแบบใช้หน่วยความจำร่วมกันแบบหลายแกน ในรายงานได้นำเสนอกลไกทั้งหมดตั้งแต่การรับข้อมูลจากผู้ใช้งานแปลงไปเป็นคิวรี่สำหรับการค้นหา การนำเข้าข้อมูลใน GPU สำหรับค้นหา และการนำผลลัพธ์การค้นหาจากการประมวลผลการค้นหาแบบขนานมาแสดง

การวัดผลด้านประสิทธิภาพมีในการค้นหาข้อมูลเชิงความหมายพบว่าใช้เวลาประมาณ 0.06 วินาทีในการค้นหาแต่ละคิวรี่ และ Speedup ที่ได้จากการใช้เครื่องที่มีจำนวน 4 แกนในการดึงข้อมูลแบบขนานประมาณ 3.6 เท่า และการใช้ MapReduce ในการรวมไฟล์ข้อมูลขนาด 4 GB ประมาณ 6-7 วินาที สำหรับการค้นหาใน GPU ด้วยวิธีการแทนค่าข้อมูลที่นำเสนอนั้น ทำให้สามารถนำเข้าข้อมูลได้มากกว่า และสามารถนำเข้าข้อมูลได้ถึง 18 ล้านเทอม ใช้เวลาค้นหา 34 มิลลิวินาที สำหรับด้านความพึงพอใจในการค้นหาเว็บเชิงความหมายจากผู้ใช้ 200 คน ได้คะแนนเฉลี่ยด้านความถูกต้องของข้อมูล และความพอใจในหน้าจอการใช้งาน 3.96/4.00

Abstract

In this project, Researchers develop the search engine for health information services in Hua Hin district. The search engine employs the parallel processing and cloud computing. The search engine connects with the health tourism data which is stored in a prototype ontology format. It supports the semantic web technology. Researchers particularly develop the search engine as a cloud application on IRIS sponsored by CAT Telecom Co. Ltd. (Thailand).

The parallel processing used in this research is two folds. First, it is for the purpose of data extraction and collection using Java Concurrent Package (1.7) together with the development of the parallel programs to implement the web crawler engine. Also, the MapReduce framework is used to combine the variety of data extraction. Secondly, the data representation for parallel query processing is designed. The parallel query processing is based on the shared-memory architecture which is on the GPU platform. The research reports the whole process from the user queries until the search results are derived and given back to the user using the representation and the parallel query.

Researchers measured the performance of the semantic query and find that it takes about 0.06 seconds to query the semantic web. Researchers measured the performance by using the parallel processing using the speedup, the speedup obtained by the parallel web crawler is about 3.6 for using 4-core computer and using MapReduce to merge the data about 4 GB takes only 6-7 seconds. The prototype of the searching in GPU can perform the search with larger data than using the search directly with the traditional RDF format. It can search 18 million terms using 34 milliseconds for searching. The user satisfaction of the semantic web development is on average 3.96/4.0 from 200 users.

สารบัญ

สารบัญภาพ

สารบัญตาราง

บทที่ 1 บทนำ

1.1 ที่มาความสำคัญ

ในแผนพัฒนาเศรษฐกิจและสังคมแห่งชาติ ฉบับที่ 11 (พ.ศ. 2554 - พ.ศ.2559) การพัฒนาศักยภาพระดับคุณภาพบริการในอุตสาหกรรมบริการและการท่องเที่ยวโดยเฉพาะจากการท่องเที่ยวเชิงสุขภาพ สามารถสร้างมูลค่าเพิ่มให้กับสาขาบริการที่เป็นมิตรกับสิ่งแวดล้อม การพำนักระยะยาวและดูแลผู้สูงอายุในท้องถิ่นด้วยความคิดสร้างสรรค์และนวัตกรรมแบบใหม่ เทคโนโลยีปัจจุบันในสาขาคอมพิวเตอร์ที่จะมีบทบาทในการปรับปรุงการให้บริการนี้ ได้แก่ การประมวลผลแบบขนานบนกลุ่มเมฆ และการบริการเว็บเชิงความหมายด้วยออนโทโลยีนั้น ซึ่งจะมีส่วนช่วยในประเด็นต่างๆ เช่น

1) ความรวดเร็วในการสืบค้นข้อมูลด้วยการประมวลผลแบบขนาน
2) คำตอบจากการสืบค้นที่ตรงตามความต้องการของผู้ค้นหาข้อมูล ด้วยการค้นหาเชิงความหมายในขอบเขตโดเมนความรู้เฉพาะทาง
3) ความสามารถในการขยายฐานข้อมูลผู้ให้บริการการท่องเที่ยวในรูปแบบต่างๆ ที่ไม่ยึดติดพื้นที่ทางกายภาพด้วยเทคโนโลยีบนกลุ่มเมฆ

ในโครงการวิจัยนี้ ผู้วิจัยขอนำเสนอ ต้นแบบระบบบริหารจัดการการท่องเที่ยวเชิงสุขภาพโดยอาศัยการผสมผสานกับเทคโนโลยีนี้สำหรับกรณีศึกษา อำเภอหัวหิน ในโครงการย่อยนี้จะเน้นการนำเสนอข้อมูลเพื่อให้สอดคล้องกับการประมวลผลแบบขนานและประมวลผลบนกลุ่มเมฆ พัฒนาวิธีการประมวลผลแบบขนานที่เหมาะสม มีประสิทธิภาพเพื่อการค้นหาข้อมูลที่รวดเร็ว

1.2 วัตถุประสงค์หลักของโครงการ

เพื่อพัฒนาระบบการค้นหาข้อมูล สำหรับให้บริการข้อมูลการท่องเที่ยวเชิงสุขภาพสำหรับอำเภอหัวหิน โดยเน้นที่การประมวลผลข้อมูลแบบขนาน และสำหรับบนกลุ่มเมฆอย่างมีประสิทธิภาพ

1.3 ขอบเขตการวิจัย

ออกแบบและพัฒนาระบบบริการข้อมูลและจัดการข้อมูลท่องเที่ยวเชิงสุขภาพสำหรับอำเภอหัวหิน โดยอาศัยเทคโนโลยีออนโทโลยี โดยระบบประกอบด้วยระบบที่รองรับการทำงานของผู้ใช้ในระดับต่างๆ ที่สามารถรวบรวมได้ พัฒนาวิธีการประมวลผลแบบขนานมาช่วยในการค้นหาข้อมูลและประมวลผลข้อมูล ประเมินประสิทธิภาพที่ประมวลผลด้านความเร็ว พัฒนาการสืบค้นเว็บเชิงความหมาย เพื่อนำรูปแบบออนโทโลยีแสดงผลการค้นหาผ่านเว็บ นำระบบบริการข้อมูลดังกล่าวไปติดตั้งทดสอบกับให้กับเทศบาลอำเภอหัวหิน

ขอบเขตพื้นที่ที่ศึกษาได้แก่ พื้นที่ทางภูมิศาสตร์ เทศบาลเมืองหัวหิน ส่วนกรณีอำเภอชะอำหรือใกล้เคียง สามารถดึงข้อมูลที่มีอยู่จากสื่ออิเล็กทรอนิกส์ได้ และตาราง 1.1 แสดงผู้ใช้งานในระบบและบทบาทของผู้ใช้

ตารางที่ 1.1 ผู้มีส่วนร่วมและบทบาท

ผู้ใช้	ระบบที่ใช้
ผู้บริหาร หัวหน้าส่วน	แสดงผลของระดับผู้บริหาร รายงานสถิติผู้เข้าใช้ รายงานสรุปความคิดเห็น เป็นต้น
ผู้ดูแลระบบ เจ้าหน้าที่ท้องถิ่น	แสดงผลของระดับผู้ดูแลระบบ การปรับปรุงข้อมูลระบบในส่วนของเหตุการณ์ (Event) ข่าวสารของเทศบาล
นักท่องเที่ยว	หน้าจอรับและแสดงผลลัพธ์จากการค้นหา เช่นข้อมูลสปา ข้อมูลสิ่งอำนวยความสะดวก บริการ ข้อมูลสถานพยาบาล ข้อมูลการอีเวนต์ ข้อมูลส่งเสริมการขาย ข้อมูลสภาพอากาศ ข้อมูลอัตราแลกเปลี่ยนประจำวัน ข้อมูลสถานที่พักแรม และอื่นๆ
ผู้ประกอบการสถานบริการ	แสดงผลข้อมูลเฉพาะสถานประกอบการ

1.4 วิธีการดำเนินการวิจัย

1.4.1 สำรวจความต้องการซอฟต์แวร์ในปัจจุบัน

1.4.2 การศึกษางานวิจัยที่เกี่ยวข้องด้านการประมวลผลแบบขนาน และกลุ่มเมฆ

1.4.3 พัฒนาตัวแบบของระบบการประมวลผลแบบกลุ่มเมฆ

1.4.4 พัฒนาอัลกอริธึมการประมวลผลคิวรี่แบบขนาน และการแทนค่าข้อมูล

1.4.5 พัฒนาโปรแกรมการประมวลผลคิวรี่แบบขนาน

1.4.6 การออกแบบหน้าจอ การติดต่อผู้ใช้ ระบบออนโทโลยี

1.4.7 ออกแบบฐานข้อมูลระบบ ได้แก่ ข้อมูลผู้ใช้ระบบ ข้อมูลสถานที่ท่องเที่ยว

1.4.8 พัฒนาระบบจัดการการค้นหา พัฒนาโปรแกรม ทดลองใช้

1.4.9 เปรียบเทียบผลการค้นหาคิวรี่ ด้านประสิทธิภาพ

1.4.10 ประเมินผลการใช้งานกับผู้ใช้บทบาทต่างๆ กัน

1.4.11 เขียนสรุปรายงานและเอกสารวิชาการ

1.4.12 จัดเตรียมการอบรมผู้ใช้ จัดทำคู่มือผู้ใช้งาน

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 ระบบจัดการและให้บริการข้อมูล เพื่อสนับสนุนการท่องเที่ยวเชิงสุขภาพ กรณีศึกษาการท่องเที่ยวเชิงสุขภาพอำเภอหัวหิน

1.5.2 อัลกอริธึม โปรแกรมการประมวลผลคิวรี่แบบขนาน และตัวแบบการประมวลผลบนกลุ่มเมฆ เพื่อสนับสนุนการท่องเที่ยวเชิงสุขภาพ กรณีศึกษาการท่องเที่ยวเชิงสุขภาพอำเภอหัวหิน

1.5.3 ผลการประเมินด้านประสิทธิภาพ ผลการประเมินด้านความถูกต้องในการค้นหา

1.5.4 คู่มือการใช้งานระบบ สรุปรายงาน เอกสารวิชาการ ผลงานตีพิมพ์

1.5.5 นักวิจัย บัณฑิตและมหาบัณฑิตในสาขาคอมพิวเตอร์ที่มีทักษะในการแก้ปัญหา

บทที่ 2 การทบทวนเอกสารและวรรณกรรมที่เกี่ยวข้อง

งานวิจัยนี้เป็นการศึกษาระบบเว็บเชิงความหมายเพื่อการจัดการและสนับสนุนการประมวลแบบขนานสำหรับการท่องเที่ยวเชิงสุขภาพ อำเภอหัวหิน โดยใช้ทฤษฎีการบริการเว็บเชิงความหมายสร้างระบบต้นแบบการเรียกใช้บริการด้านการท่องเที่ยว และการประมวลผลแบบขนานเพื่อเพิ่มประสิทธิภาพของการรับส่งข้อมูลออนโทโลยี การประมวลผลฐานความรู้บนกลุ่มเมฆ การคิวรี่ข้อมูล Triple และการค้นหาฐานความรู้ออนโทโลยีในสถานที่ท่องเที่ยวจริง เพื่อให้สามารถประเมินประสิทธิภาพของการใช้ทรัพยากรทั้งหมด ในการค้นหาฐานความรู้ด้านการท่องเที่ยวเชิงสุขภาพที่มีการเชื่อมต่อของข้อมูลทั่วโลกที่มีการปรับปรุงสม่ำเสมอจากลิงค์เดต้า (Linked Data Cloud) นอกจากนี้ยังมีระบบการบริหารจัดการท่องเที่ยวสำหรับกลุ่มผู้ใช้ที่เป็นนักท่องเที่ยว ผู้ดูแลระบบ ผู้บริหารและนักเทคนิค

จึงขอนำเสนอวรรณกรรมที่เกี่ยวข้องกับการสร้างระบบโดยรวม นอกเหนือจากที่กล่าวถึงในแผนงานวิจัย ยังมีงานวิจัยที่เกี่ยวข้องกับการบริการกลุ่มเมฆ แพลตฟอร์มเกี่ยวกับงานบริการเกี่ยวกับท่องเที่ยว และการประมวลแบบขนาน รวมทั้งพื้นฐานด้านการประมวลผลบนกลุ่มเมฆ (Cloud Computing) การบริการเว็บเชิงความหมาย (Semantic Web Service) และการประมวลผลแบบขนาน (Parallel Processing)

2.1 งานวิจัยเกี่ยวกับการบริการบนกลุ่มเมฆ

บริการจากเว็บเซอร์วิสจากอเมซอน (Amazon) เป็นมาตรฐานที่ได้รับการยอมรับในกลุ่ม Infrastructure as a service (IaaS) ที่ผู้ใช้มีการจ่ายค่าบริการตามการใช้งานผ่าน VM Elastic Compute Cloud (EC2) [1] ซึ่งถือว่าอเมซอน เป็นผู้นำตลาดในด้าน Xen-based virtualized infrastructure[2] ในด้านการตลาด อเมซอนมักจะจัดการส่งเสริมการขายด้วยการลดราคา นอกจากนี้อเมซอนยังสร้างพันธมิตรในกลุ่มเมฆสาธารณะในเรื่องการรักษาสิ่งแวดล้อมที่ได้รับการรับรองเรื่องความปลอดภัยและนำเสนอที่เก็บข้อมูลส่วนกลางที่ครอบคลุมทั่วโลก

ในการประมวลผลกลุ่มเมฆที่มีลักษณะเป็นศูนย์ข้อมูล (Data Center) เช่นนี้ ผู้ให้บริการต้องคุณลักษณะสำคัญ [1] ดังนี้

1) มีเจ้าของเป็นหน่วยงานต่อไปนี้ เช่น องค์กรหรือบริษัท ผู้ให้บริการ (Service Provider) หรือรัฐบาล

2) สามารถสร้างกลุ่มของทรัพยากรที่ใช้ในการประมวลผลทั้งหมด และบริการต่างๆ ที่ต้องใช้ร่วมกันและเปิดให้ผู้ใช้สมัครใช้งานได้

---

[1] http://aws.amazon.com/ec2/

[2] http://www.networkworld.com/supp/2012/enterprise2/040912-ecs-iaas-companies-257611.html

3) มีการเก็บค่าบริการทั้งทรัพยากรและบริการ ตามการใช้งานจริงและ/หรือตามขนาดของแพจเกจบริการพื้นฐาน

4) มีการให้บริการกระจายตามภูมิภาคของโลก ซึ่งในกรณีนี้เป็นการสร้างความโปร่งใสต่อกลุ่มผู้ใช้งาน

5) มีการเพิ่มสิ่งที่เตรียมการและการตั้งค่า และถอนออกได้โดยอัตโนมัติ ทั้งทรัพยากรและบริการที่อยู่บนการบริการพื้นฐานแบบช่วยตัวเองของระบบ โดยปกติจะเปิดให้ผู้ใช้ใช้ได้หลังการสมัครอยู่แล้ว ซึ่งเป็นการตั้งค่าของโปรแกรมที่ไม่ต้องมีมนุษย์เป็นผู้ควบคุมและส่งออกมาเป็นลำดับในเวลาไม่กี่วินาที

6) ทั้งทรัพยากรและการบริการถูกส่งออกมาในลักษณะเสมือนจริง ทั้งเซิร์ฟเวอร์ ดิสก์ และอุปกรณ์เครือข่ายซึ่งล้วนแต่เป็นการใช้ทรัพยากรเสมือนจริง โดยดึงจากโครงสร้างพื้นฐานทางกายภาพที่ผู้ใช้ไม่จำเป็นต้องมองเห็นได้

7) โครงสร้างทางกายภาพควรมีการเปลี่ยนแปลงน้อยมาก แต่ทรัพยากรและการบริการเสมือนนั้นสามารถเปลี่ยนได้ทันที

8) ทรัพยากรและบริการนั้นเป็นลักษณะกายภาพเสมือน ทั้งเซิร์ฟเวอร์ ดิสก์ และอุปกรณ์เครือข่าย หรือในส่วนนามธรรม ทั้งฟังก์ชันการจัดเก็บวัตถุ (blob storage) ฟังก์ชันคิวข้อความ ฟังก์ชันอีเมล์ ฟังก์ชันมัลติคาสต์ ทั้งหมดนี้สามารถเข้าถึงได้โดยการรันโค้ดหรือสคริปต์เพื่อเซ็ตของ API สำหรับบริการเหล่านี้ หรืออาจใช้ทั้งโค้ดและสคริปต์ผสมกัน

ตัวอย่างสำหรับการบริการ Amazon Elastic Compute Cloud (EC2) มีรูปแบบต่างๆ ดังนี้

1) 750 ชั่วโมงสำหรับ Amazon EC2 Linux (ยกเว้น SuSe Enterprise Server และ RHEL) Micro Instance usage หน่วยความจำ ขนาด 613 MB และรองรับ 32 บิต และ 64 บิต โดยมีจำนวนชั่วโมงรันอย่างต่อเนื่องในแต่ละเดือน *

2) 750 ชั่วโมงสำหรับ Amazon EC2 Microsoft Windows Server‡ Micro Instance usage

หน่วยความจำ ขนาด 613 MB และรองรับ 32 บิต และ 64 บิต โดยมีจำนวนชั่วโมงรันอย่างต่อเนื่องในแต่ละเดือน *

3) 750 ชั่วโมงสำหรับan Elastic Load Balancer และการประมวลผลข้อมูลขนาด 15 GB *

4) 30 GB ชั่วโมงสำหรับ Amazon Elastic Block Storage และ จำนวน I/O ขนาด 2 ล้านครั้ง และมีพื้นที่ 1 GB snapshot storage*

ข้อยกเว้นมีดังนี้

* เฉพาะลูกค้าใหม่ที่เปิดใช้บริการ เมื่อครบ 12 เดือนแล้วจะจ่ายตามการใช้งานจริง

สำหรับการจ่ายตามจริงมีนโยบายดังต่อไปนี้ กรณีที่เลือกใช้แหล่งทรัพยากรที่ใกล้ที่สุดคือสิงคโปร์และเลือกชนิดการใช้งาน (on demand) ซึ่งเหมาะกับโครงการระยะเวลาไม่นาน (เช่นไม่เกิน 1 ปี) มีข้อดีคือไม่ต้องเสียค่าธรรมเนียมเบื้องต้น สามารถวางแผนการ การจัดซื้อและบำรุงรักษาฮาร์ดแวร์ที่มีความยืดหยุ่นสูงได้ รวมถึงการย้ายระบบด้วยต้นทุนที่น้อยกว่า ราคาในตาราง 2.1 นั้นสามารถใช้ได้ทั้งกิจการสาธารณะและส่วนบุคคลบน

ระบบปฏิบัติการเฉพาะ ซึ่งในกรณีวินโดว์ คือ Windows Server® 2003 R2, 2008 และ 2008 R2 นอกจากนี้ยังใช้วินโดว์กับ SQL Server, Amazon EC2 บน SUSE Linux Enterprise Server, Amazon Red Hat Enterprise Linux และ Amazon EC2 บน IBM ซึ่งการคิดราคาจะแตกต่างไป สามารถดูเพิ่มเติมได้ที่ http://aws.amazon.com/ec2/#pricing

ทั้งนี้ผลิตภัณฑ์ของอเมซอน หรือใน Google App Engine[3] ล้วนมีคุณสมบัติเหล่านี้ทั้งสิ้น

ตัวอย่างของการบริการของอเมซอนที่มีการส่งเสริมการขายสำหรับลูกค้าใหม่เป็นเวลา 1 ปีดังนี้ AWS Free Usage Tier (ต่อเดือน)[4]

ตารางที่ 2.1 ค่าใช้จ่ายกับความคุ้มค่าในการลงทุนต่อประมาณการใช้งาน[5]

	การใช้งานระบบ Linux/UNIX	**การใช้งานในระบบ Windows**
แพคเกจมาตรฐาน (Standard On-Demand Instances)		
Small (Default)	$0.085 ต่อชั่วโมง	$0.115 ต่อชั่วโมง
Medium	$0.170 ต่อชั่วโมง	$0.230 ต่อชั่วโมง
Large	$0.340 ต่อชั่วโมง	$0.460 ต่อชั่วโมง
Extra Large	$0.680 ต่อชั่วโมง	$0.920 ต่อชั่วโมง
แพคเกจตามความต้องการขนาดเล็ก (Micro On-Demand Instances)		
Micro	$0.020 ต่อชั่วโมง	$0.020 ต่อชั่วโมง
แพคเกจตามความต้องการขนาดใหญ่ (High-Memory On-Demand Instances)		
Extra Large	$0.506 ต่อชั่วโมง	$0.570 ต่อชั่วโมง
Double Extra Large	$1.012 ต่อชั่วโมง	$1.140 ต่อชั่วโมง
Quadruple Extra Large	$2.024 ต่อชั่วโมง	$2.280 ต่อชั่วโมง
แพคเกจตามความต้องการใช้ซีพียู (High-CPU On-Demand Instances)		

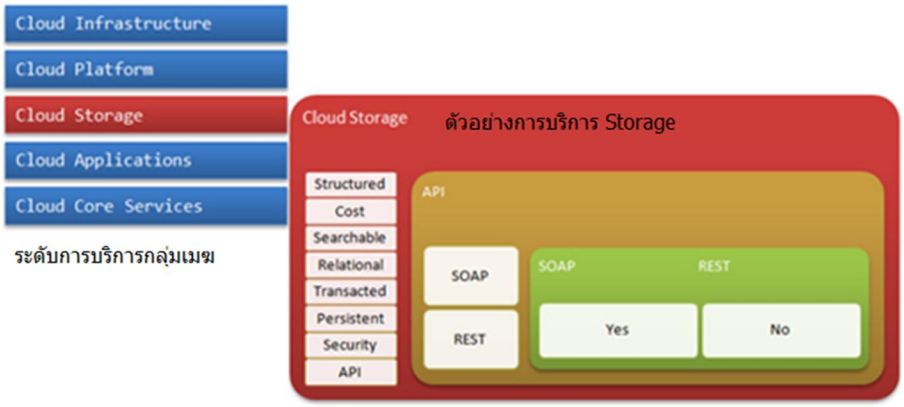
[3] http://code.google.com/appengine/, [Online] Access: 17 Aug 2012.

[4] http://aws.amazon.com/free/

[5] http://aws.amazon.com/ec2/#pricing

Medium	$0.186 ต่อชั่วโมง	$0.285 ต่อชั่วโมง
Extra Large	$0.744 ต่อชั่วโมง	$1.140 ต่อชั่วโมง
แพคเกจตามความต้องการใช้คลัสเตอร์ (Cluster Compute Instances)		
Quadruple Extra Large	N/A*	N/A*
แพคเกจตามความต้องการใช้จีพียู (Cluster GPU Instances)		
Quadruple Extra Large	N/A*	N/A*
* Cluster and High I/O Instances are not available in all regions		

สถาปัตยกรรมของแพลตฟอร์มนั้นเป็นส่วนหนึ่งที่สำคัญ ในการย้ายแอพพลิเคชันไปบนระบบประมวลผลกลุ่มเมฆ[6] มี 2 วิธีในการพิจารณาเลือกแพลตฟอร์ม วิธีแรกคือ การเลือกคุณสมบัติของแอพพลิเคชันให้เหมาะกับกลุ่มเมฆ เพื่อตรวจสอบดูว่าบริการของกลุ่มเมฆเหมาะสมกับแอพพลิเคชันนั้นๆ หรือไม่ จากนั้นจึงกำหนดประเภทของบริการที่ใช้ วิธีที่ 2 คือการประเมินบริการจากผู้ให้บริการ โดยเป็นการประเมินความเหมาะสมของบริการในการเป็นโฮสต์ของแอพพลิเคชัน เพื่อจะได้เลือกบริการที่เข้ากันได้ และอาจมีการพัฒนาเพิ่มเติมต่อไป



ภาพที่ 2.1 การบริการของกลุ่มเมฆ 5 ระดับและตัวอย่างการบริการ Storage

(ที่มา: http://i.msdn.microsoft.com/dynimg/IC263304.jpg, [Online], Access: Aug 17, 2012.)

ภาพที่ 2.1 เป็นตัวอย่างการให้บริการเก็บข้อมูลจากบริการกลุ่มเมฆของบริษัทยไมโครซอฟต์ ซึ่งผู้ใช้ต้องพิจารณาความเหมาะสมของบริการที่เลือกใช้ ระดับการบริการมีหลายระดับตั้งแต่ระดับ Infrastructure ไปจนถึง Core Service ในรูปด้านขวาเป็นการเลือกบริการระดับ Storage ซึ่งจะมีการสนับสนุนการบริการเว็บใน

[6] http://msdn.microsoft.com/en-us/library/dd430340, [Online] Access: 17 Aug 2012.

รูปแบบ SOAP[7] แต่ไม่รองรับ REST[8] และมีการรองรับด้านความปลอดภัย การทำงานแบบ Transaction ฐานข้อมูลสัมพันธ์ เป็นต้น

Martens [2] ทำการวัดบริการกลุ่มเมฆแบ่งโดยเป็น โบรกเกอร์ สถานที่ ประเภทบริการและทรัพยากร Service-level Agreement (SLA) และระบบรักษาความปลอดภัย สำหรับการวัดด้วย KPI ประกอบด้วย ชนิดและการทำงาน การวัดความเสี่ยงนั้นพิจารณาที่ความคิดเห็นต่อความเสี่ยง ผลกระทบในภาพรวม และประเภทความเสี่ยง และมุมมองด้านความร่วมมือที่ประกอบด้วย ระดับความร่วมมือ การตรวจสอบความร่วมมือ ประเภทของการประมวลผลข้อมูลและประเภทข้อบังคับด้านความร่วมมือ

---

7 http://en.wikipedia.org/wiki/SOAP

8 http://en.wikipedia.org/wiki/Representational_state_transfer

2.2 แพลตฟอร์มสำหรับการท่องเที่ยวทางอิเล็กทรอนิกส์

งานบริการด้านการท่องเที่ยวอิเล็กทรอนิกส์ (e-Tourism) สามารถนำไปทำงานบนกลุ่มเมฆได้ ในงานวิจัย [3] ได้เสนอแบบงานบริการที่เกี่ยวกับการท่องเที่ยวเป็นงานย่อย ประกอบด้วย แพ็กเกจแบบพลวัตร (dynamic packaging) การวางแผนการท่องเที่ยว การเปรียบเทียบราคา การออกแบบเส้นทางการเดินทาง และการโฆษณาทางการตลาดและการส่งเสริมการขาย ซึ่งสามารถได้แบ่งตามสถานการณ์การทำงาน ตามภาพที่ 2.2 ที่ได้แสดงโครงงานการท่องเที่ยวอิเล็กทรอนิกส์ ที่แบ่งขั้นตอนการทำงานที่สำคัญดังนี้

ขั้นตอนที่ 1: นักท่องเที่ยวส่งคำร้องไปยังตัวแทนการท่องเที่ยวเพื่อให้ตัวแทนการท่องเที่ยวทำการวางแผนการท่องเที่ยวโดยใช้องค์การเสมือน (Virtual Organization) ทำให้สามารถใช้งานข้ามองค์กรได้ เป็นการเปิดสิทธิ์การบริการแก่การท่องเที่ยวแบบหนึ่ง

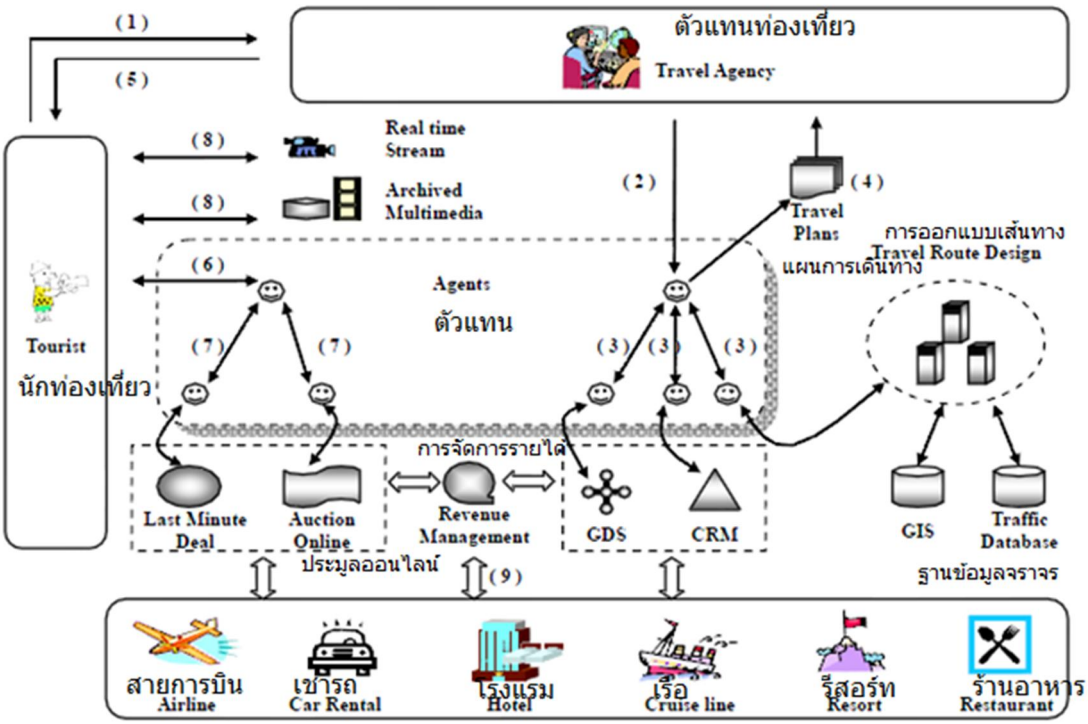
ขั้นตอนที่ 2: ตัวแทนการท่องเที่ยวส่งแผนการท่องเที่ยวให้แก่เอเจนต์

ขั้นตอนที่ 3: ตัวแทนติดต่อตัวแทนอื่นๆ และกำหนดงานแก่สาขาย่อย งานของสาขาแรก คือ ออกแบบเส้นทางการท่องเที่ยว ซึ่งใช้การคำนวณที่ซับซ้อนตามฐานข้อมูลสารสนเทศทางภูมิศาสตร์ (GIS) งานของอีกสองสาขาคือการคิวรี่ไปที่กรมอุตุนิยมวิทยาเพื่อขอผลการพยากรณ์อากาศ และอัตราค่าที่พักจากหน่วยงาน Global Distribution Framework (GDS) และ Central Reservation Management (CRM)

ขั้นตอนที่ 4: ตัวแทนแรกประมวลผลจากแต่ละพันธมิตรทางธุรกิจและส่งแผนการท่องเที่ยวมาให้ตัวแทนการท่องเที่ยว

ขั้นตอนที่ 5: ตัวแทนการท่องเที่ยวส่งแผนการท่องเที่ยวให้แก่นักท่องเที่ยว

จากขั้นตอนการทำงานของการท่องเที่ยวอิเล็กทรอนิกส์ข้างต้นนี้ สามารถนำไปประยุกต์ใช้กับโครงสร้างในอนาคตได้ ซึ่งประกอบด้วย เอเจนต์ส่วนบุคคลและเอเจนต์อัจริยะ การบริการข้อมูลเชิงความหมาย การให้ลำดับการเลือกโดยสนับสนุนการตัดสินใจของมนุษย์ การเชื่อมต่อหลายระบบให้ราบรื่น การประมวลผลระหว่างระบบแบบกระจาย การเก็บข้อมูลและความร่วมมือระหว่างพันธมิตร ซึ่งสามารถนำมาเรียบเรียงตามระดับชั้นการส่งข้อมูลได้ดังตารางที่ 2.2



ภาพที่ 2.2 โครงงานของการท่องเที่ยวอิเล็กทรอนิกส์ [3]

ตารางที่ 2.2 กรอบการทำงานของการท่องเที่ยวเชิงอิเล็กทรอนิกส์โดยรวม

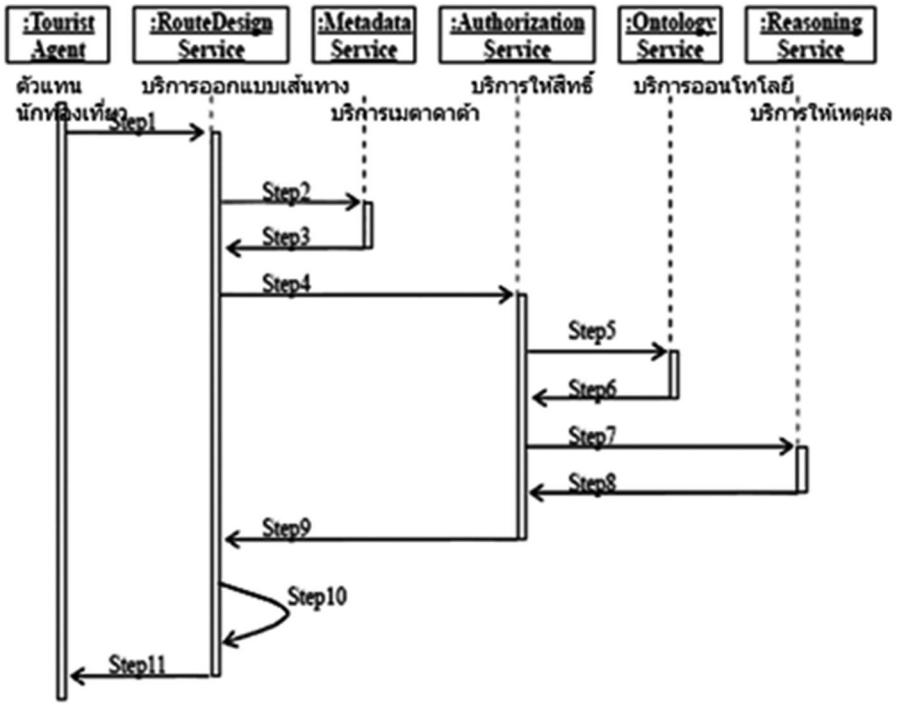
ระดับชั้น	การบริการ	หน้าที่
มุมมองการบริการ	-ผู้บริโภค -ผู้ให้บริการ -องค์กรเสมือน	นักท่องเที่ยวและตัวแทนการท่องเที่ยวใช้บริการ ตัวแทนการท่องเที่ยวและตัวแทนธุรกิจ โรงแรม สนามบิน เรือสำราญ และอื่นๆ ใช้บริการ สนับสนุนการบริการส่วนผู้บริโภค
มุมมองเนื้อหา	-ความรู้	ช่วยจัดการความรู้ระหว่างชั้นเทคนิคและงานบริการผู้ใช้ส่วนผู้ให้บริการ โดยประมวลความรู้มาจากข้อมูลและสารสนเทศที่ดึงมาจากแหล่งเว็บแบบกระจายในชั้นเทคนิค
มุมมองเทคโนโลยี	-โครงสร้างระบบบนกลุ่มเมฆ - ตัวแทนอัจฉริยะ - เว็บเชิงความหมายและบริการ	โครงสร้างระบบบนกลุ่มเมฆ ซึ่งดัดแปลงมาจากระบบกริด [1] ส่วนนี้ช่วยสนับสนุนการทำงานระหว่างองค์กรแบบเปิดและมีมาตรฐานมากขึ้น สนับสนุนการการประมวลผลระหว่างระบบแบบกระจาย การเก็บข้อมูลและจัดการการไหลของงาน ระหว่างตัวแทนอัจฉริยะ และเว็บเชิงความหมายและการบริการ ในชั้นเทคนิคมีการเชื่อมต่อกันและเชื่อมกับบริการความรู้ (Knowledge service) ในชั้นเนื้อหา มีหน้าที่หลักคือการเชื่อมต่อระบบอย่างแนบเนียนและทำการอนุมานความรู้ในระดับโมเดลและอินสแตนซ์

ในส่วนโครงสร้างออนโทโลยีประกอบด้วยส่วนแนวคิด (Concepts) บริการ (Services) ทรัพยากร (Resources) และบทบาท (Roles) โดยมีหน้าที่โดยละเอียดดังนี้

- **แนวคิด (Concepts ontology)** ทำการกำหนดคำมาตรฐานตามที่ WTO (World Tourism Organization) กำหนดไว้ซึ่งมีความเป็นสากล [5] และเป็นศัพท์ตามมาตรฐานแล้ว
- **บริการ (Services ontology**) กำหนดหน้าจอการบริการ และฟังก์ชันตามมาตรฐาน OTA [9] ตัวอย่างระดับชั้นของการบริการการท่องเที่ยวอาทิ กรณีที่แบ่งเป็นประเภทบริการพาหนะและคมนาคม และประเภทของบริการทัวร์ที่แบ่งโดยละเอียดเป็นทัวร์ที่นำเสนอโดยตรง และบริการการค้นหาตามความสนใจคือบริการการค้นหาด้านการท่องเที่ยว อาทิจุดมุ่งหมายของการค้นหาแบ่งเป็นขนาดของข้อมูล (ขนาดมากที่สุดหรือน้อยที่สุด) และประเภทของตำแหน่งที่ตั้ง
- **ทรัพยากร (Resources ontology)** กล่าวถึงความสามารถของฮาร์ดแวร์ ซอฟต์แวร์ และการเชื่อมต่อที่สนับสนุนบริการต่างๆ ที่เปิดให้เรียกใช้ ตัวอย่างของการบริการฮาร์ดแวร์คือความสามารถของหน่วยประมวลผลกลางและความเร็วของเครือข่าย

[9] http://www.opentravel.org/specifications/default.aspx

- **บทบาท (Roles ontology)** ประกอบด้วยการอธิบายบทบาทของผู้ใช้ คือนักท่องเที่ยว ตัวแทนการท่องเที่ยวและผู้ให้บริการต่างๆ กระบวนการให้สิทธิ์ตามหน้าที่เป็นดังภาพที่ 2.3 กระบวนการให้สิทธิ์ใน Role Ontology



ภาพที่ 2.3 กระบวนการให้สิทธิ์ใน Role Ontology [3]

ภาพที่ 2.3 แสดงกระบวนการให้สิทธิ์ใน Role ontology ซึ่งประกอบด้วย 10 ขั้นตอน คือ ขั้นตอนที่ 1 นักท่องเที่ยวเข้าสู่องค์กรเสมือนเพื่อขอ*บริการออกแบบเส้นทาง*การเดินทางผ่านตัวแทนส่วนตัว *บริการออกแบบเส้นทาง* ทำหน้าที่เก็บคุณสมบัติของนักท่องเที่ยวตามบริการเมทาดาทาในขั้นตอนที่ 2 และ 3 ถัดมาขั้นตอนที่ 4 *บริการออกแบบเส้นทาง*สร้างการร้องขอสิทธิที่มีคุณสมบัติตามฐานความรู้ RDF ตามจุดให้บริการลูกค้า ในขั้นตอนที่ 5 และ 6 *บริการให้สิทธิ์*ได้รับออนโทโลยี VO ที่ประกอบด้วยนิยามของ Role จาก *ออนโทโลยีบริการ* และในขั้นที่ 7 และ 8 *บริการให้สิทธิ์*จะไปนำ*บริการทางเหตุผล*มาอนุมานหน้าที่ของนักท่องเที่ยวผ่านออนโทโลยี องค์กรเสมือนและพารามิเตอร์คุณสมบัติของนักท่องเที่ยว จากนั้น*บริการให้สิทธิ์*เปรียบเทียบหน้าที่ของนักท่องเที่ยวที่อนุมานได้กับหน้าที่การเข้าถึงพื้นฐานเพื่อประเมินว่านักท่องเที่ยวใช้บริการใดได้บ้าง ในขั้นตอนที่ 9 ถ้าอนุญาตให้เข้าถึงได้ไปขั้นตอนที่ 10 และส่งค่าเส้นทางการท่องเที่ยวแก่ตัวแทนของนักท่องเที่ยว ถ้ามีปฏิเสธการเข้าถึง จะไม่คำนวณเส้นทางและตัวแทนของนักท่องเที่ยวจะได้รับข้อความปฏิเสธในการเข้าถึง

วิธีการติดตามความสนใจของนักท่องเที่ยว การวิเคราะห์ความชอบและแสดงผลอัตโนมัติ เป็นการกำหนดค่าที่สนใจไว้ในอะเรย์ IV_{ij} *(i=1...n เป็นจำนวนสิ่งที่สนใจ, j=0, 1)* ถูกใช้โดยตัวแทนเพื่อแสดงระดับความสนใจของนักท่องเที่ยว

เมื่อ j=0, IVij={สิ่งที่สนใจ};

เมื่อ j=1, IVij={1,2,3,4,5}.

ระดับความสนใจ: กำหนดเป็นค่าดังนี้

1: ถ้าจำนวนบริการในเดือนน้อยกว่า 5

2: ถ้าจำนวนบริการในเดือนมากกว่า 5 แต่น้อยกว่า

3: ถ้าจำนวนบริการในเดือนมากกว่า 10 แต่น้อยกว่า 15

4: ถ้าจำนวนบริการในเดือนมากกว่า 15 แต่น้อยกว่า 20

5: ถ้าจำนวนบริการในเดือนมากกว่า 20

จากค่าที่กำหนดข้างต้นสิ่งที่สนใจคือเซตที่ได้มาจากการร้องขอจากข้อมูลของนักท่องเที่ยวในเดือนที่แล้ว *ระดับความสนใจ* คือค่าความถี่ของการร้องขอมาจากข้อมูลของนักท่องเที่ยวในเดือนที่แล้ว ถ้านักท่องเที่ยวร้องขอสิ่งนั้นมาก ค่าระดับความสนใจนี้จะมีมาก เช่นถ้ามีการร้องขอ ดังนี้ “Request (op: airFare, source: Beijing, destination: New York)” เป็นการร้องขอข้อมูลราคาตั๋วเครื่องบินจากปักกิ่งไปยังนิวยอร์ค ถ้าความถี่ในการร้องขอนี้ มีค่ามากกว่าระดับที่กำหนด การร้องขอนี้จะเข้าไปอยู่ในรายชื่อเส้นทางโปรด ในการเข้าใช้ระบบครั้งหน้าระบบก็จะแสดงเส้นทางบินจากปักกิ่งไปนิวยอร์คของนักท่องเที่ยว

นักวิจัย Lee [6] ได้ทำการวิเคราะห์ค่าขอบเขต (bound) ของการท่องเที่ยวเชิงสุขภาพด้านการแพทย์ที่มีประเทศสิงคโปร์เป็นประเทศผู้นำอยู่ โดยทำการวิเคราะห์ค่าตัวแปรที่สำคัญ 2 ตัว คือ TOUR ซึ่งเป็นความต้องการของนักท่องเที่ยวนำมาจาก Singapore Tourism Board (STB) และ HEALTH เป็นจำนวนผู้ให้บริการทางการแพทย์ต่อประชากร 10,000 คนที่ได้มาจากกระทรวงสาธารณสุข (Ministry of Health) ของประเทศสิงคโปร์ ประโยชน์ที่ได้จะช่วยในการวิเคราะห์การลงทุนด้านบุคลากรของรัฐบาลในปัจจุบันให้เพียงพอต่อนักท่องเที่ยวที่เข้ามาในระยะยาว การทดสอบค่า bound ต่างๆ เป็นดังตาราง 2.3

ตารางที่ 2.3 วิเคราะห์ค่า bound ของการท่องเที่ยวเชิงสุขภาพด้านการแพทย์ที่ประเทศสิงคโปร์

การนำไปใช้	สูตร
พิจารณาจากความสัมพันธ์ระยะยาวระหว่าง 2 ตัวแปร คือ TOUR และ HEALTH แสดงด้วยโมเดล unrestricted error correction	$\Delta \mathrm{TOUR}_t = a_0 + \sum_{i=1}^{p} a_{\mathrm{Ti}} \Delta \mathrm{TOUR}_{t-i} + \sum_{i=0}^{p} a_{Hi}\, \Delta \mathrm{HEALTH}_{t-i} + a_1 \mathrm{TOUR}_{t-1} + a_2 \mathrm{HEALTH}_{t-1} + \varepsilon_{1t}$ $\Delta \mathrm{HEALTH}_t = b_0 + \sum_{i=1}^{p} b_{\mathrm{Hi}}\, \Delta \mathrm{HEALTH}_{t-i} + \sum_{i=0}^{p} b_{\mathrm{Ti}}\, \Delta \mathrm{TOUR}_{t-i} + b_1 \mathrm{HEALTH}_{t-1} + b_2 \mathrm{TOUR}_{t-1} + \varepsilon_{2t}$
ค่าฐานของ bound testing มาจาก ARDL (Autoregressive distributed lag) ซึ่งสามารถสร้างโมเดล ARDL ตามสัมประสิทธิ์โดยประมาณระยะยาว	$(1 - c_1 L - \ldots - c_e L^e)\mathrm{TOUR}_t = \mathrm{d}_0 + \left(1 - \mathrm{d}_1 L - \ldots - \mathrm{d}_f L^f\right)\mathrm{HEALTH}_t + \varepsilon_{3t}$ โดยค่า TOUR สามารถนำมาปรับพารามิเตอร์ตามค่า residual ในภายหลังได้
Causality test ตามงานของ Granger ที่ Lee [4] กล่าวอ้าง ผลลัพธ์คือค่า short-run (F-test) และ long-run (t-test)	$\Delta \mathrm{TOUR}_t = g_0 + \sum_{i=1}^{p} g_{\mathrm{Ti}}\, \Delta \mathrm{TOUR}_{t-i} + \sum_{i=1}^{p} g_{\mathrm{Hi}}\, \Delta \mathrm{HEALTH}_{t-i} + \gamma \mathrm{ECT}_{t-1} + \varepsilon_{4t}$ $\Delta \mathrm{HEALTH}_t = h_0 + \sum_{i=1}^{p} h_{\mathrm{hi}}\, \Delta \mathrm{HEALTH}_{t-i} + \sum_{i=1}^{p} h_{\mathrm{Ti}}\, \Delta \mathrm{TOUR}_{t-i} + \varepsilon_{5t}$

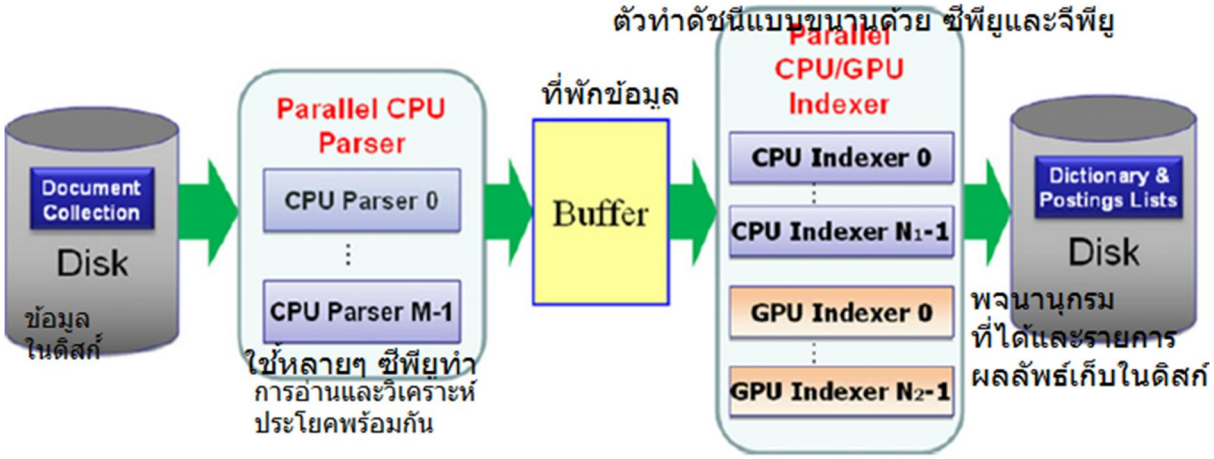
2.3 การคำนวณแบบขนานด้วย MapReduce

การทำงานในรูปแบบ MapReduce เป็นการสร้างโมเดลสำหรับการคำนวณแบบขนานสำหรับระบบงานแบบกระจายสำหรับข้อมูลปริมาณมาก ประกอบด้วยขั้นตอนการแมบ (Map) โดยจะทำการแมบงานย่อยไปยัง worker ต่างๆ เพื่อแก้ปัญหาแล้วผลลัพธ์ชั่วคราวที่ได้จะอยู่ในรูป <key, value> และนำออกมาส่งต่อให้ worker เพื่อทำการรีดิวซ์ (Reduce) ซึ่งเป็นการรวบผลลัพธ์เข้าด้วยกัน โดยโหนดมาสเตอร์จะนำ <key, value> มาจับกลุ่มอัตโนมัติ โดยข้อมูลที่มี key เดียวกันจะเข้ามาอยู่รวมกลุ่มกัน

มีงานวิจัยเป็นจำนวนมากที่ทำงานแบบขนานในขั้นตอนการค้นหาข้อมูลนี้ เพื่อส่งคำตอบกลับไปยังผู้ใช้งาน และยังมีการประมวลผลดัชนีการค้นหาด้วยวิธี MapReduce โดยตรง ในการทำดัชนีจะใช้เทอม (term) ที่กำหนดเป็น key และ ID ของเอกสารเป็น value ซึ่ง worker ที่ทำการ Reduce จะรับรายชื่อ ID ที่ยังไม่ได้เรียงลำดับมาโดยตรง และได้มีงานวิจัยที่ใช้ MapReduce แบบที่ทำในรอบเดียว (Single-Pass) มาปรับปรุง โดย McCreadie และคณะวิจัยใน [6] ได้นำเสนอให้ Map worker ส่ง <term, partial posting list> แทน เพื่อลดจำนวนการส่งและขนาดของการส่งทั้งหมดระหว่างขั้นตอนการ Map และการ Reduce เนื่องจากการรวมเทอมที่ซ้ำนั้น จะลดความถี่ในการส่งลง กลยุทธ์นี้ทำให้ความเร็วของการประมวลผลเพิ่มขึ้นเมื่อจำนวนของคอร์ของเครื่องมีมากขึ้น

Lin และคณะ [7] ได้พัฒนาวิธี MapReduce แบบ scalable MapReduce Indexing โดยการเปลี่ยน <term, posting{document ID, term frequency}> เป็น <tuple{term, document ID}, term frequency> การทำเช่นนี้ส่งผลส่วนให้ key นั้นไม่ซ้ำ และรับรองได้ว่า MapReduce ส่งค่าไปยังขั้น Reduce ตามลำดับ ส่งผลให้ลิสต์ที่ส่งไปเรียงกันโดยไม่เหลือผลลัพธ์ให้รอไปทำยังรอบต่อไป ซึ่งการทำงานแบบนี้ Lin และ Dyer [7] ได้เผยแพร่เป็นหนังสืออัลกอริธึมสำหรับ MapReduce ที่รู้จักกันดีว่ามีอัตรา throughput ที่เหมาะสำหรับการทำดัชนีของข้อความแบบเต็ม (full text)

ยังมีงานวิจัยเกี่ยวกับการประมวลผลแบบขนานด้วยอัลกอริธึม MapReduce ในขั้นตอนการทำดัชนี ได้ใช้หน่วยประมวลผลกลางและหน่วยประมวลผลกราฟิกส์แบบผสม [8] อัลกอริธึมของการทำดัชนีแบบนี้ยังเป็นการลดรายการข้อมูลที่เก็บลงชั่วคราว ค่า document ID ของเอกสารถูกเก็บแบบเรียงลำดับอยู่ในแต่ละรายการที่ต้องใช้ทำงานต่อไป ดังนั้นแนวคิดอย่างง่ายคือจะทำการเข้ารหัสที่ช่องว่างระหว่าง document ID ของเอกสารที่ติดกัน 2 รายการแทนการรวมค่าตามการเข้ารหัสแบบไบต์ (byte encoding) และแบบ γ และการบีบอัดแบบ Golomb



ภาพที่ 2.4 การไหลของข้อมูลสำหรับการทำดัชนีด้วยวิธีการแบบขนาน [10]

ภาพที่ 2.4 เป็นการนำหน่วยประมวลผลกราฟิกส์หรือจีพียู (GPU) มาช่วยหน่วยประมวลผลกลางหรือซีพียู (CPU) ในการประมวลผล โดยได้มีการทดสอบแล้วว่า การนำหน่วยประมวลผลกราฟิกส์เข้ามาช่วยประมวลผล ทำให้สามารถลดต้นทุนและเวลาได้ วิธีการก็คือทำการแบ่งข้อมูลและส่งไปวิเคราะห์ประโยค (parse) พร้อมกันเป็นลักษณะ Trie-Collection นำเข้าไปทำงานในหน่วยประมวลผลกลางและในหน่วยประมวลผลกราฟิก จากการทดลองพบว่าการนำหน่วยประมวลผลกราฟิกส์เข้ามาช่วยส่งผลให้ประสิทธิภาพการทำงานดีกว่า [9,10]

นอกจากการสร้างดัชนีเพื่อจัดการแสดงผลแก่ผู้ใช้แล้วยังมีการทำเหมืองเว็บ (Web mining) ด้วยเว็บเชิงความหมายเพื่อแสดงคำตอบแก่ผู้ใช้อย่างเหมาะสม รวมทั้งการแปลงดัชนีแบบอื่น [9] นอกจากนี้ยังมีงานวิจัยที่การศึกษาขั้นตอนการทำงานแบบขนานของวิธีการ MapReduce ในกระบวนการอื่นที่เกี่ยวข้อง [10] ทำการพัฒนาวิธีการประมวลผลแบบขนานที่รองรับการค้นหาข้อมูลดังกล่าวที่อาศัยหลักการแปลงและส่งข้อมูลขนาดใหญ่ผ่านเครือข่ายประยุกต์ใช้บิตเมตริกซ์ของ G.T. Williams, Medha Atre และคณะ [12] และใช้หลักการของ MapReduce แบ่งการจัดการเป็นการประมวลผลแบบขนานนั้น อ้างอิงหลักการการคิวรี่แบบขนานของRDF/OWL บน MapReduce [13] การประมวลผลด้วยภาษา SPARQL สำหรับการทำงานบนกลุ่มเมฆ [14] และการเก็บข้อมูลแบบ Triple สำหรับ MapReduce [15] การปรับปรุงมิดเดิลแวร์เพื่อควบคุมการทำงานของหน่วยประมวลผลกราฟิก [16] อัลกอริธึมการทำงานแบบขนานในส่วนของกฎและข้อมูลสำหรับออนโทโลยี [17] ที่กล่าวถึงในแผนงานวิจัยแล้วและเทคนิคการกระจายข้อมูลสำหรับแอพพลิเคชันแบบขนานในลักษณะโฮตส์ และไคลเอนท์ [18]

ในงานวิจัยของ Hung-chih Yang, Ali Dasdan, Ruey-Lung Hsiao และ D. Stott Parker [19] เป็นการเพิ่มการทำงานในส่วนของการรวมกันในการทำ MapReduce จากเดิมที่ไม่สนับสนุนการประมวลผลสำหรับข้อมูลที่ต่างกัน จึงได้ทำการพัฒนาตัวแบบขึ้นมาใหม่ ให้มีประสิทธิภาพในการรวมข้อมูลที่ถูกแบ่งเป็นส่วนๆ หรือถูกแบ่งประเภทไว้ ซึ่งนำมาใช้กับการค้นหากับข้อมูลจำนวนมหาศาล

ในที่นี้รูปแบบ MapReduce นั้นสืบทอดมาจาก Google MapReduce ยกเว้นแต่คุณสมบัติย่อยๆ เท่านั้นที่เปลี่ยนแปลง ที่เพิ่มเข้าไปได้แก่ ฟังก์ชันการรวม (merge) ฟังก์ชัน processor ฟังก์ชันการแบ่ง (partition selector) และการทำซ้ำแบบปรับได้ (configurable iterator) ในส่วนการรวม ผู้ใช้อาจต้องการเหตุผลในการประมวลผลข้อมูลต่างๆ ไป ทั้งนี้ขึ้นอยู่กับข้อมูลด้วย หลังจากงานของ Map และ Reduce ทำเสร็จแล้ว จะเรียกการทำการรวมไปยังคลัสเตอร์แต่ละโหนดเช่นเดียวกันกับ MapReduce ที่ต้องพิจารณาเหตุผลในการประมวลผลข้อมูลตามข้อมูลที่เข้ามา หน้าที่ฟังกชันต่างๆ ได้แก่

- **Partition selector**: ในการรวม ให้ผู้ใช้กำหนดฟังก์ชัน partition selector เพื่อมากำหนดข้อมูลที่แบ่งแล้ว ซึ่งผลิตโดย up-stream reducers จะถูกเรียกไปยังการรวมหรือ merge ต่อ
- **Processor function**: ตัว processor ถูกวางอยู่ในที่ที่ผู้ใช้สามารถกำหนดตรรกะของการประมวลผลข้อมูลจากแต่ละแหล่งที่มา
- **Merge function**: ผู้ใช้สามารถกำหนดตรรกะของการประมวลผลข้อมูล ในการรวมข้อมูลจาก 2 แหล่งที่มา ที่ซึ่งข้อมูลเหล่านี้เหมาะสมกับเงื่อนไขการรวมได้
- **Configurable iterator**: เพื่อจัดการความสัมพันธ์ของตัว merger ของ logical iterator จำนวน 2 ตัว ผู้ใช้สามารถระบุอัลกอริธึมการรวม ต่างๆ เองได้

Djoerd Hiemstra และ Claudia Hauff [20] กล่าวถึงการค้นหาข้อมูลจากเอกสารหน้าเว็บที่เก็บรวบรวมไว้ ได้แก่ ClueWeb09 ด้วยการใช้ MapReduce บน Hadoop ช่วยในการดึงข้อมูลขนาดใหญ่นี้ โดยระบบจะสแกนเอกสารทั้งหมดและอ่านแต่ละหน้าเว็บเพียงครั้งเดียว เพื่อสกัดข้อความออกมา ทั้งยังมีการเก็บสถิติคำศัพท์ที่มีอยู่ในเอกสารเหล่านั้น แล้วทำการค้นหาแบบเชิงเส้น จากนั้นจะลบหน้าเว็บที่ไม่เกี่ยวข้องออกไป การทดลองแบ่งเป็น 4 ประเภท คือ การค้นหาแบบธรรมดา การค้นหาแบบบอกหัวเรื่อง การค้นหาแบบลบหน้าเว็บที่ไม่เกี่ยวข้องออกไป สุดท้ายการค้นหาแบบบอกหัวเรื่องและลบหน้าเว็บที่ไม่เกี่ยวข้องออกไป ผลการทดลองพบว่าแต่ละประเภทการค้นหามีค่าความแม่นยำ (Precision) เพิ่มขึ้นตามลำดับ แสดงให้เห็นว่ามีจำนวนเอกสารที่เกี่ยวข้องกับความต้องการและถูกสกัดออกมาได้มากจากจำนวนเอกสารที่มีอยู่ทั้งหมด เนื่องจากระบบสามารถกำจัดเอกสารที่ไม่เกี่ยวข้องออกไปได้เป็นจำนวนมาก แต่ถ้ามีจำนวนเอกสารเพิ่มมากขึ้นโดยแบ่งประเภทการค้นหาตามรูปแบบเดิม พบว่ามีค่าความแม่นยำลดลงตามลำดับ เนื่องจากมีจำนวนเอกสารที่ไม่ได้เกี่ยวข้องมากนั่นเอง

Jeffrey Dean และ Sanjay Ghemawat [21] นำเสนอการประยุกต์การประมวลผลแบบ MapReduce บนกลุ่มคอมพิวเตอร์ขนาดใหญ่ อธิบายการทำงานและยกตัวอย่างโปรแกรม โดยสรุปได้ดังนี้ MapReduce เป็นรูปแบบการเขียนโปรแกรมและการดำเนินการต่างๆ ที่เกี่ยวข้องกับการประมวลผลและสร้างชุดข้อมูลขนาดใหญ่ ซึ่งทำงานแบบขนานได้อย่างอัตโนมัติ สามารถคำนวณข้อมูลได้ปริมาณมาก โดยแบ่งข้อมูลเป็นกลุ่มเล็กๆ ทั้งยังจัดเวลาการทำงานเมื่อมีการกระทำข้ามเครื่อง ดูแลการสื่อสารระหว่างคอมพิวเตอร์และจัดการข้อผิดพลาดที่เกิดขึ้น วิธีการนี้ช่วยให้โปรแกรมเมอร์ที่ไม่มีประสบการณ์กับระบบแบบขนานและแบบกระจายเขียนโปรแกรมได้ง่ายขึ้น เพราะมีการซ่อนรายละเอียดของการทำงานภายในต่างๆไว้ วิธีการนี้ได้รับแรงบันดาลใจจากวิธี Map และ Reduce ของการทำงานในหลายๆ ภาษาโปรแกรม

การทำงานของ MapReduce แบ่งเป็น 2 ส่วน คือ Map และ Reduce โดยการทำงานของ Map จะแบ่งข้อมูลที่ได้รับมาเป็นขนาดเล็กๆ ข้อมูลที่ถูกแบ่งได้รับการประมวลผลจากคอมพิวเตอร์เครื่องต่างๆในเครือข่าย โดยมีโหนดหลัก (master) เป็นคอมพิวเตอร์เครื่องหนึ่งทำหน้าที่กำหนดและควบคุมการทำงานทั้งหมดของเครื่องคอมพิวเตอร์ที่เหลือซึ่งเรียกว่า โหนดทำงาน (worker หรือ slave) เมื่อประมวลผลเสร็จแล้วจึงเก็บข้อมูลลงบัฟเฟอร์ในดิสก์หรือหน่วยความจำของเครื่องตัวเอง จากนั้นการทำงานของ Reduce ก็จะมาอ่านข้อมูลในบัฟเฟอร์แล้วจัดกลุ่มข้อมูล โดยโหนดหลักจะเก็บสถานะของการทำงาน Map และ Reduce คือ ว่าง (idle) กำลังทำงาน (in-progress) และทำงานเสร็จสมบูรณ์ (complete) โหนดหลักมีการติดต่อกับโหนดทำงานเป็นระยะๆ ถ้าหากไม่มีการตอบสนองกลับมาก็จะจดจำว่าโหนดทำงานนั้นล้มเหลว และเปลี่ยนสถานะโหนดนั้นให้เป็นว่าง

การที่มีข้อจำกัดด้านทรัพยากรเครือข่าย MapReduce จึงเก็บข้อมูลนำเข้าไว้ในหน่วยความจำของเครื่องตัวเอง ทำให้ลดปริมาณข้อมูลที่ต้องส่งผ่านเครือข่าย และเพื่อความสะดวกในการเข้าถึงและค้นหาข้อมูลอีกด้วย และด้วยภาระงานของ Map และ Reduce ที่แตกต่างกัน จึงต้องมีการจัดสมดุลของภาระงานแบบพลวัต (Dynamic load balance) เพื่อให้การทำงานของแต่ละเครื่องมีประสิทธิภาพสูงสุด และสามารถกู้คืนได้เร็วหากเกิดข้อผิดพลาด

การทำงานแบบ MapReduce สนับสนุนการอ่านและเขียนข้อมูลได้หลายรูปแบบ โดยแฟ้มข้อมูลแต่ละประเภทจะรู้ด้วยตัวเองว่าควรแบ่งแฟ้มข้อมูลอย่างไรเพื่อให้สื่อความหมายในการประมวลผล ขณะที่โหนดหลักทำงานจะมีการรายงานสถานะของข้อมูล คือ ความก้าวหน้าของการคำนวณ เช่น จำนวนงานที่เสร็จแล้ว จำนวนงานที่กำลังประมวลผล หรือขนาดของข้อมูล เพื่อให้ผู้ใช้ทำนายเวลาที่ใช้ในการทำงานได้ รวมทั้งตัดสินใจได้ว่าควรเพิ่มทรัพยากรอะไรให้กับระบบเพื่อการคำนวณบ้าง เช่น รายงานการใช้ซีพียู การใช้ดิสก์ รายงานแบนด์วิดธ์ของเครือข่าย

การทำงานที่สามารถใช้การคำนวณแบบ MapReduce ได้เช่นโปรแกรม Grep ที่มีการสแกนและค้นหาข้อความจากเอกสารขนาดใหญ่ โดยแบ่งข้อมูลออกเป็นกลุ่ม ผลการคำนวณพบว่าอัตราที่ข้อมูลถูกสแกนสูงขึ้นเมื่อจำนวนเครื่องคอมพิวเตอร์มากขึ้น งานที่สำคัญอีกชิ้นหนึ่งคือ สร้างระบบดัชนี ซึ่งผลิตโครงสร้างข้อมูลสำหรับการบริการค้นหาหน้าเว็บของ Google เป็นต้น

นอกจากนี้ยังมีการงานวิจัยใน [22] เน้นการจัดสรรทรัพยากรแบบทำซ้ำ (Iterative resource allocation) สำหรับหน่วยความจำ สำหรับอัลกอริธึมการค้นหาแบบขนานบนกลุ่มเมฆ กริด และคลัสเตอร์ที่ใช้งานร่วมกัน อัลกอริธึมแบบ iterative allocation นี้ทำงานซ้ำจนกว่าจะสามารถแก้ปัญหาได้เสร็จสิ้น จะเห็นได้ว่าอัลกอริธึมนี้สามารถทำงานได้ง่าย แต่มีส่วนสำคัญคือ ฟังก์ชัน *Increase()* ซึ่งรับค่าของจำนวนของ HAUs เพื่อจัดสรรพื้นที่หน่วยความจำในรอบต่อไป ดังอัลกอริธึมต่อไปนี้

Algorithm 1 Generic, Iterative Allocation (IA)

```
numHAUs ← 1
while true do
    result ← Run algorithm a on problem p
            if result = solved then
            return success
          else
           numHAUs ← Increase(numHAUs)
```

2.4 งานวิจัยที่เกี่ยวข้องด้านการประมวลผลกับข้อมูลขนาดใหญ่

การประมวลผลแบบขนานกับข้อมูลปริมาณมากในปัจจุบัน ส่วนใหญ่จะใช้อัลกอริธึม MapReduce [21] เป็นพื้นฐาน โดยมีการแบ่งข้อมูลออกเป็นแฟ้มข้อมูลก่อน จากนั้นจึงจับคู่แฟ้มข้อมูลที่มีปัญหาแบบเดียวกันเข้าด้วยกัน เมื่อแก้ปัญหาเสร็จแล้วจึงนำคำตอบที่ได้มารวมกัน จากตารางที่ 2.4 แสดงถึงงานวิจัยที่มีการประมวลผลแบบขนานบนกลุ่มเมฆ โดย Rohloff และ Schantz [23] ใช้กรอบการทำงานแบบ MapReduce แล้วสามารถทำการประมวลผลข้อมูลขนาดใหญ่บนกลุ่มเมฆได้อย่างรวดเร็วร่วมกับการวางแผนการประมวลกฎ ขณะที่การประมวลผลข้อมูลของ Zhang [3] มีลักษณะเป็นการประยุกต์ใช้กริดที่เน้นในขอบเขตของการท่องเที่ยว ดังตารางที่ 2.4

ตารางที่ 2.4 การประมวลผลแบบขนานบนกริดและบนกลุ่มเมฆ

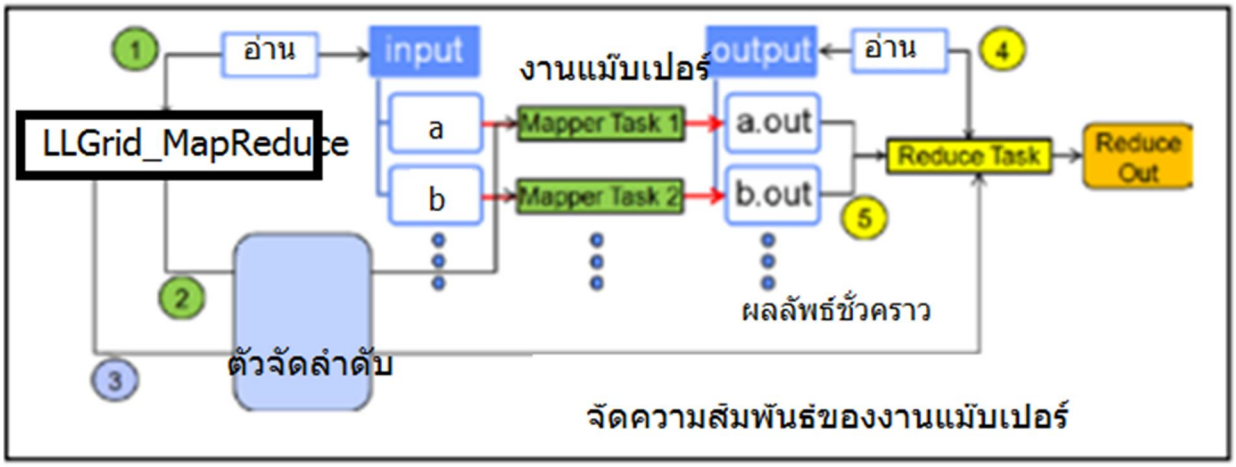
งานวิจัย	ตัวอย่างที่ใช้	สิ่งแวดล้อม	วิธีการ	สรุปผล
Rohloff และ Schantz [15,23]	14 คิวรี่ (LUBM) 6000 มหาวิทยาลัย (ประมาณ 800 Triple)	ใช้ Hadoop บน Cloudera บริการบนกลุ่มเมฆของ Amazon EC2 ปริมาณโหนดคำนวณ 20 XL	ใช้กรอบการทำงานแบบ MapReduce บนระบบคลาวด์สำหรับระดับล่างโดยใช้อินพุต RDF แบบแฟ้มข้อมูล N3	ระบบกระจายการประมวลผลขนาดใหญ่ที่มีประสิทธิภาพสูงโดยใช้กรอบการทำงานแบบ MapReduce เป็นซอฟต์แวร์ใช้การเก็บข้อมูล แบบ Triple ชื่อ SHARD ในระยะสั้นและเพื่อการปรับปรุงในระยะยาว ใช้การแคชของข้อมูลในหน่วยความจำและสำรองข้อมูลบนดิสก์ในระดับซอฟต์แวร์
Zhang[3]	เตรียมแต่ละชั้นของบริการกริดเพื่อโดเมนการท่องเที่ยวอิเล็กทรอนิกส์	บริการกริดสองชั้น คือชั้นของส่วนประกอบพื้นฐานและชั้นแอพพลิเคชันแพลตฟอร์มของกริดคือ Globus 4 และมิดเดิลแวร์คือ Jena	ชั้นส่วนประกอบพื้นฐานเป็นไปตาม S-OGSA ใช้กรอบการทำงานของ OntoGrid ชั้นแอพพลิเคชันใช้บริการการท่องเที่ยวอิเล็กทรอนิกส์เช่น RouteDesign	นำเสนอกริดเชิงความหมายเพื่อโดเมนการท่องเที่ยวอิเล็กทรอนิกส์สำหรับเว็บเซอร์วิสใช้ Apache Axis และ WSRF เพื่อประกาศและประมวลการร้องขอ SOAP การออกแบบออนโทโลยีเป็นการท่องเที่ยวอิเล็กทรอนิกส์แบบ OTA-Compliant ของ TourSearchRQService

สำหรับงานวิจัยที่มีโครงสร้างข้อมูลและอัลกอริธึมแบบขนานนั้น ขึ้นอยู่กับระบบสถาปัตยกรรมระบบและฮาร์ดแวร์ด้วยเช่นกัน จากตารางที่ 2.5 นำเสนองานวิจัยที่มีการประมวลผลข้อมูลแบบข้อความเช่นเดียวกันแต่ Lin และคณะ [24] ได้ใช้ขั้นตอนวิธีแบบ MapReduce บนหน่วยประมวลผลกลาง แต่ Zheng และ JaJa [8] นำข้อมูลชุดเดียวกันมาจัดรูปแบบใหม่และนำหน่วยประมวลผลกราฟิกมาช่วยในการประมวลผลร่วมปรากฏว่าสามารถทำงานได้มากกว่าในเวลาที่น้อยกว่า และได้ทดลองวิธีเดียวกันกับฐานข้อมูลอื่นด้วย

ตารางที่ 2.5 โครงสร้างข้อมูลและอัลกอริธึมแบบขนานบนฮาร์ดแวร์ต่างๆ

ผู้วิจัย	ตัวอย่าง	**สิ่งแวดล้อม**	**วิธีการ**	สรุปผล
Lin และคณะ [24]	เซตข้อมูลเป็น ClueWeb09	ใช้ MapReduce แบบ Ivory จำนวน 99 โหนด แต่ละโหนดมีสองแกนรวม 198 แกนบนคลัสเตอร์หน่วยประมวลผลกลาง	การประมวลผลข้อความด้วยการทำดัชนีข้อความทั้งหมดและขยายขนาดการใช้ของอัลกอริธึม MapReduce [8,9 in project II] มี Throughput 91.96 MB ต่อวินาที	สร้างรายการโพสต์ตำแหน่งโดยอัลกอริธึม MapReduceแบบ Ivory ซึ่งยังมีการคิดค่าใช้จ่ายบางส่วนอยู่
Zheng และ JaJa [8]	เซตข้อมูลเป็น ClueWeb09 และ Wikipedia01-07	ใช้ CPU และ GPU ใน 1 โหนดประกอบด้วยหน่วยประมวลผลกลาง 8 แกนและหน่วยประมวลผลกราฟิกส์จำนวน 2 ตัว) ผลลัพธ์ที่ดีสุดเกิดจากใช้หน่วยประมวลผลกลาง 6 แกนวิเคราะห์คำและ 2 แกนทำดัชนีร่วมกับหน่วยประมวลผลกราฟิกส์อีกสองตัว	สร้างชุด Index แบบไทร์ (trie) จาก 0 – 17612 มี throughput 262.76 MB/s ให้ผลดีกว่า Lin [24]	ให้เอกสารจากดิสก์หมายเลข 0 เป็นเอกสารนำเข้าชั่วคราวจากการวิเคราะห์ ถือเป็นขั้นตอนเตรียมการเพื่อทำดัชนีแบบขนานระหว่างซีพียู และหน่วยประมวลผลกราฟิกส์เป็นขั้นเตรียมการของการทำรายการในดิสก์หมายเลข 1 เช่นกัน

สำหรับวิธีการประมวลผลและภาษาที่คล้ายคลึงกับงานวิจัยนี้ได้แก่งานเรื่องการประมวลผลข้อมูลขนาดใหญ่ด้วยคลัสเตอร์ MPI (Message Passing Interface[10]) Byun และคณะ [25] ได้มีการนำเสนออัลกอริธึม MapReduce กับงานใดๆ โดยการทำงานเริ่มจากการสแกนแฟ้มข้อมูลอินพุตจากไดเรคทอรี่หรืออ่านจากรายการในแฟ้มอินพุตชั้นที่ 1 แล้วจึงเข้าถึงกระบวนการตั้งตารางงานในชั้นที่ 2 โดยการสร้างตารางงานจากหลายๆ งานที่เกิดขึ้น และเรียกตารางนี้ว่าอะเรย์ของงาน แล้วตั้งชื่อให้เป็น งานแมพเพอร์ (Mapper task) ที่ 1 งานแมพเพอร์ที่ 2 ไปเรื่อยๆ ซึ่งในการนี้อาจใช้โปรแกรมช่วยให้มีการทำงานที่พร้อมกัน (concurrent) เมื่ออะเรย์ของงานถูกสร้างและส่งไปประมวลผลแล้ว แต่ละแฟ้มข้อมูลที่เป็นอินพุตจะถูกประมวลผลโดยงานหนึ่งๆ ที่กำหนดไว้ตามคำสั่งเรียกว่าแมพเพอร์ในภาพที่ 2.5 สำหรับในส่วนนี้นั้นแอพพลิเคชันที่สร้างขึ้นสามารถประมวลในรูปแบบใดก็ได้เช่น ทำงานบนเชลสคริปต์ (Shellscript) ทำงานด้วยภาษาจาวาหรือทำงานด้วยโปรแกรมใดๆ ที่เขียนด้วยภาษาต่างๆ



ภาพที่ 2.5 การทำงานของ LLGrid MapReduce

เมื่ออ่านแฟ้มอินพุตทั้งหมดแล้ว จะแจกงานไปยังงานแม๊บเปอร์ ตัวจัดลำดับมาควบคุมลำดับการทำงานของงานแม๊ปเปอร์อีกที เมื่อแต่ละแม๊ปเปอร์ได้ผลลัพธ์ชั่วคราวแล้ว จะส่งไปงาน Reduce เพื่อรวบรวมในขั้นตอนสุดท้าย

พารามิเตอร์ที่เกี่ยวข้องในงานนี้จึงประกอบด้วย จำนวนของงาน วิธีการ Map วิธีการ Reduce ไดเรคทอรี่ของอินพุตและเอาต์พุต งานวิจัยนี้นำวิธีการนี้มาประยุกต์ใช้ด้วยการนำข้อมูลจากเว็บไซต์มาประมวลผลแบบขนานก่อนในการดึงข้อมูลขึ้นมาเพื่อใส่ลงในออนโทโลยีต่อไป

2.5 งานวิจัยที่เกี่ยวข้องด้านการแทนค่าข้อมูล

ปัจจุบันมีระบบการเก็บข้อมูลแบบ Triple เป็นจำนวนมาก และมีการใช้โครงสร้างข้อมูลแบบกราฟหรือต้นไม้ ช่วยให้สามารถใส่ข้อมูล Tripleได้มากขึ้น ตัวอย่างของระบบ Triple ที่เป็นที่รู้จัก เช่น[11] AllegroGraph

---

[10] http://en.wikipedia.org/wiki/Message_Passing_Interface

[11] http://www.w3.org/wiki/LargeTripleStores

(ความจุ 1+Trillion) OpenLink Virtuoso v6.1 (ความจุ 15.4Billion+ explicit; uncounted virtual/inferred) BigOWLIM (12B explicit, 20B total) โดยคิดค่าบริการ 100,000 คิวรี่ต่อ $1 Garlik 4store (15 Billion) Bigdata® (12.7Billion) YARS2 (7 Billion) Jena TDB (1.7Billion) Jena SDB (650 Million) Mulgara (500 Million) RDF gateway (262 Million) Jena with PostgreSQL (200 Million) Kowari (160 Million) 3store ด้วย MySQL 3 (100 Million) และ Sesame (70Million) เป็นต้น ในรายชื่อเหล่านี้มี OWLIM ที่สามารถใช้กับ Amazon EC2 ได้โดยคิดค่าบริการ 1 US$ ต่อการคิวรี่ SPARQL จำนวน 100,000 คิวรี่

สำหรับกรณีซอฟต์แวร์ Open source มีระบบเก็บฐานความรู้แบบ Triple ที่ทำงานบน Hadoop โดยใช้โครงสร้างข้อมูลกราฟได้แก่ SHARD [26] โดยที่การออกแบบ SHARD ประกอบด้วยฐานข้อมูลแบบ Triple ที่อยู่บน Hadoop File System (HDFS) ที่มีเมธอดเรียกจากไคลเอนท์ ประมวลผลบนกลุ่มเมฆด้วยงานจาก MapReduce ขยายขนาดได้ รองรับการทำงานของคิวรี่ SPARQL และมีการอนุมานขั้นระดับพื้นฐาน และในงานวิจัย [23] ได้นำเสนออัลกอริธึม Clause-Iteration ทำงานบนกราฟโดยใช้วิธีการแบบ MapReduce เช่นกัน

2.6 การประมวลผลบนกลุ่มเมฆกับการบริการเว็บเชิงความหมาย

ตารางที่ 2.6 นำเสนอทฤษฎีการประมวลผลบนกลุ่มเมฆ ตามระดับชั้นของการบริการ จะเห็นว่าในการประมวลบนกลุ่มเมฆมีระดับชั้น 3 ระดับได้แก่ Software as a Service, Platform as a Service และ Infrastructure as a Service ในแต่ละชั้นสอดคล้องโครงสร้างบริการของเว็บเชิงความหมายดังนี้

ตารางที่ 2.6 ทฤษฎีที่เกี่ยวข้องตามระดับชั้นของการประมวลผลบนกลุ่มเมฆ

การประมวลผลบนกลุ่มเมฆ	**การบริการแบบเว็บเชิงความหมาย**
Software as a Service (SaaS)	เว็บแอพพลิเคชัน HTML5 บริการเว็บเซอร์วิส ของผู้ให้บริการต่างๆ เช่นสายการบิน ตัวแทนท่องเที่ยว API เพื่อการส่งออกไปยังเครื่องมือต่างๆ
Platform as a Service (PaaS)	ชั้นตัวกลางของ Semantic Web (เช่น Reasoner, Inference, Query) ตัวจัดการออโนโลยี (Ontology Manager) (XML syntax: RDF, OWL) Web Annotation แอพพลิเคชัน Java Runtime
Infrastructure as a Service (IaaS)	เครื่องเซิรฟ์เวอร์ เครือข่าย ที่จัดเก็บออนโทโลยี

โดยระบบเว็บเชิงความหมายจะมีความแตกต่างกับระบบที่เป็นอยู่ในปัจจุบัน สรุปได้ดังตาราง 2.7

ตารางที่ 2.7 ความแตกต่างและความได้เปรียบของระบบปัจจุบันกับระบบเว็บเชิงความหมายที่ใช้อยู่

ประเด็นปัญหา	**การออกแบบเว็บในปัจจุบันที่มีอยู่ ตัวอย่างคือเว็บของเทศบาล เว็บการท่องเที่ยวแห่งประเทศไทย และเว็บแนะนำที่พักแรม**	**เว็บเชิงความหมายจากงานวิจัย มีลักษณะคือเว็บที่ตอบผลการค้นหาจากผู้ใช้ตามโครงสร้างออนโทโลยี**
การค้นหาและนำเสนอแบบหลายมิติ	ค้นหาโดยใช้ตัวหนังสือและมีลำดับของการนำเสนอที่ไม่สัมพันธ์กันจำนวนมาก เช่น ผลการค้นหาคำว่า Cicada ใน Google Search Engine สามารถแสดงผลทางกีฏวิทยาได้ด้วยแต่ความตั้งใจของผู้ค้นหาคือสถานที่ในหัวหิน	ค้นหาตามโครงสร้างออนโทโลยีที่มีความสัมพันธ์ตามโดเมนที่สร้างหลายมิติ เช่น การค้นหาคำว่า Cicada เมื่ออยู่ในโดเมนอำเภอหัวหินแล้วเป็นชื่อตลาด อยู่ที่ตำบลหนองแก
ลักษณะการนำเสนอของเว็บไซต์	นำเสนอแบบแสดงข้อความที่ผู้สร้างต้องการแสดง ซึ่งมีความหมายเพื่อมนุษย์อ่านและเข้าใจ เบื้องหลังมีฐานข้อมูลที่เป็นข้อความเหมือนกันเมื่อเปิดอ่าน มนุษย์ก็สามารถเข้าใจได้ แต่เครื่องจักรไม่เข้าใจสิ่งเหล่านี้	นำเสนอแบบแสดงข้อความที่ผู้สร้างต้องการแสดง ซึ่งมีความหมายเพื่อมนุษย์อ่านและเข้าใจ แต่เบื้องหลังระดับเครื่องสามารถทำให้เครื่องเข้าใจกันได้โดยใช้โครงสร้างออนโทโลยีแบบ XML และเครื่องต้น-ปลายทางสามารถแลกเปลี่ยนบริการกันได้ ตัวอย่างเช่น การรับข้อมูลที่มีการปรับอัตโนมัติ ข่าวสาร RDF จากเหตุการณ์ต่างๆ
วิวัฒนาการของการขยายตัวของโครงสร้างข่าวสาร	ข่าวสารถูกเก็บและบันทึกในที่เดียวกัน การออกแบบสร้างเพื่อดึงข้อมูลจากฐานข้อมูลเดียว แบบบนลงล่างและบำรุงรักษาจากส่วนกลาง	ข่าวสารถูกออกแบบเพื่อเชื่อมต่อกับแหล่งอื่นจึงเป็นแบบขยายกิ่งโครงสร้าง โดยพัฒนาจากล่างขึ้นบน และมีการอัพเดตแบบกระจายได้
การขยายโครงสร้างของชุมชนการสื่อสาร	ชุมชนเพิ่มข่าวสารและแสดงผลตามโครงสร้างของเว็บ เช่นเว็บบอร์ด	ชุมชนสามารถเพิ่มชั้นและโครงสร้างใหม่และเพิ่มข่าวสารตามโครงสร้างนั้นได้ เช่นออนโทโลยีที่ออกแบบเหมือนกันสามารถนำโครงสร้างมาใช้ใหม่ หรือนำออนโทโลยีโดเมนการท่องเที่ยวในชุมชนอื่นมาเชื่อมกันได้ ตัวอย่างเช่นการขยายโครงสร้างของลิงค์เดต้า (linkeddata.org) นั่นคือใช้โครงสร้างเดียวกันสามารถขยายโดเมนการท่องเที่ยวระดับอำเภอเป็นทั่วประเทศได้และขยายไปสัมพันธ์กับโดเมนอื่นที่เกี่ยวข้องกันได้
การขยายมุมมองของชุมชนการสื่อสาร	การจัดเก็บเนื้อหาและการจัดการอยู่ที่ส่วนกลาง	จัดการและจัดเก็บเนื้อหาข้อมูลแบบกระจายได้ แต่อยู่ในมุมมองรูปแบบเดียวกัน

ประเด็นปัญหา	การออกแบบเว็บในปัจจุบันที่มีอยู่ ตัวอย่างคือเว็บเทศบาล เว็บการท่องเที่ยวแห่งประเทศไทย และเว็บแนะนำที่พักแรม	เว็บเชิงความหมายจากงานวิจัย มีลักษณะคือเว็บที่ตอบผลการค้นหาจากผู้ใช้ตามโครงสร้างออนโทโลยี
การรวมทรัพยากรที่กระจายอยู่	ผู้ดูแลระบบต้องหาข้อมูลของแต่ละเว็บ แต่ละชุดเอกสาร เพื่อปรับปรุงทีละหน้าผ่านฟอร์ม เมื่อต้องการบำรุงรักษาต้องทำทีละสำเนา	เนื่องจากผู้พัฒนาผลิตโครงสร้างและข้อมูลที่นำกลับมาใช้ใหม่ได้ ผู้ดูแลระบบสามารถอัพเดตข้อมูลที่เดียวแล้วกระจายข้อมูลหลายเว็บเพจได้ เช่นจุดบริการจอดรถใหม่สามารถเพิ่มคุณสมบัติการค้นหาสถานที่ใกล้เคียง เพื่อเชื่อมต่อกับสถานที่ๆ ตั้งบนถนนใกล้กันได้
การรวมข่าวสารข้ามเว็บ	เป้าหมายเว็บอยู่ที่การเข้าชมของคน เมื่อต้องการใช้ข้อมูลร่วมกันจะเป็นลิงค์เชื่อมไปหา เช่นลิงค์แหล่งท่องเที่ยวจากเว็บเทศบาลไปยังเว็บอื่นๆ	เมื่อมีการอนุญาตระหว่างเจ้าของเว็บเครื่องสามารถสื่อสารเพื่อเข้าถึงข่าวสารโครงสร้างเดียวกันผ่านบริการได้ เช่นการดึงข้อมูลราคาของสถานที่พักแรมจากเว็บชื่อดังมาเปรียบเทียบกัน

2.7 เทคโนโลยีของ Hadoop

ปัจจุบันปัญหาเกี่ยวกับการจัดการข้อมูลขนาดใหญ่ กำลังได้รับความสนใจเป็นอย่างมากทั้งทางภาคธุรกิจและทางด้านงานวิจัย เนื่องจากข้อมูลต่างๆ ที่จัดเก็บมีความสำคัญ เพื่อใช้ในการสืบค้น วิเคราะห์ รวมทั้งประมวลผลข้อมูล อีกทั้งข้อมูลเหล่านั้นมีความลำบากยุ่งยากมากขึ้น ดังนั้นปัญหาที่สำคัญในการจัดการกับข้อมูลขนาดใหญ่ คือ การเก็บข้อมูลและการค้นหาข้อมูล เพื่อแก้ปัญหาในการจัดเก็บข้อมูลขนาดใหญ่ จึงได้มีการนำเทคโนโลยีที่เรียกว่า Hadoop เข้ามาช่วย

Hadoop เป็นซอฟต์แวร์เฟรมเวิร์ค (Framework) ที่ถูกออกแบบมาเพื่อทำงานบนระบบคอมพิวเตอร์แบบกระจาย และสนันสนุนการทำงานแบบขนาน โดยมีชุดคำสั่ง API เพื่อช่วยอำนวยความสะดวกแก่นักพัฒนาแอพพลิเคชันที่จะสร้างระบบค้นหาหรือวิเคราะห์ข้อมูลขนาดใหญ่ อาจกล่าวได้ว่า Hadoop เป็นส่วนพื้นฐานที่ทำหน้าที่ทั้งประมวลผลแบบขนาน และจัดเก็บข้อมูลแบบกระจาย

Hadoop ใช้สำหรับทำงานกับแอพพลิเคชันบนคลัสเตอร์ของคอมพิวเตอร์ขนาดใหญ่ ซึ่งเหมาะสำหรับแอพพลิเคชันที่ต้องการความน่าเชื่อถือ โดย Hadoop ถูกสร้างจากแบบจำลองการคำนวณคือ Map Reduce ซึ่งจะทำการแบ่งแอพพลิเคชันออกเป็นชิ้นงานเล็กๆ โดยชิ้นงานแต่ละชิ้นอาจถูกทำซ้ำ โดยโหนดหลักจากนั้นก็จะทำการรวบรวมงานที่ถูกกระจายออกไปในแต่ละโหนดลูกกลับเข้ามาเป็นชิ้นเดียว นอกจากนี้ Hadoop ยังมีระบบแฟ้มแบบกระจาย (Distributed File System) ซึ่งเรียกว่า Hadoop Distributed File System (HDFS) โดยจะรองรับการส่งรับข้อมูลจำนวนมากระหว่างคลัสเตอร์ และจัดเก็บแฟ้มข้อมูลขนานใหญ่ที่มีความเชื่อถือได้ ซึ่งมีผลต่อการจัดการอินพุตเอาท์พุต และแบนด์วิดธ์ โดย HDFS จะทำการสำรองข้อมูลบนโหนดคำนวณ (compute node) และเตรียมช่องทางการสื่อสารไว้สำหรับการส่งข้อมูลข้างคลัสเตอร์ ซึ่งทั้ง Map/Reduce และ HDFS ได้ถูกออกแบบมาเพื่อให้เป็นเฟรมเวิร์กที่สามารถจัดการกับโหนดที่ทำงานล้มเหลวหรือไม่เสถียรสามารถทำงานต่อไปได้โดยอัตโนมัติด้วย

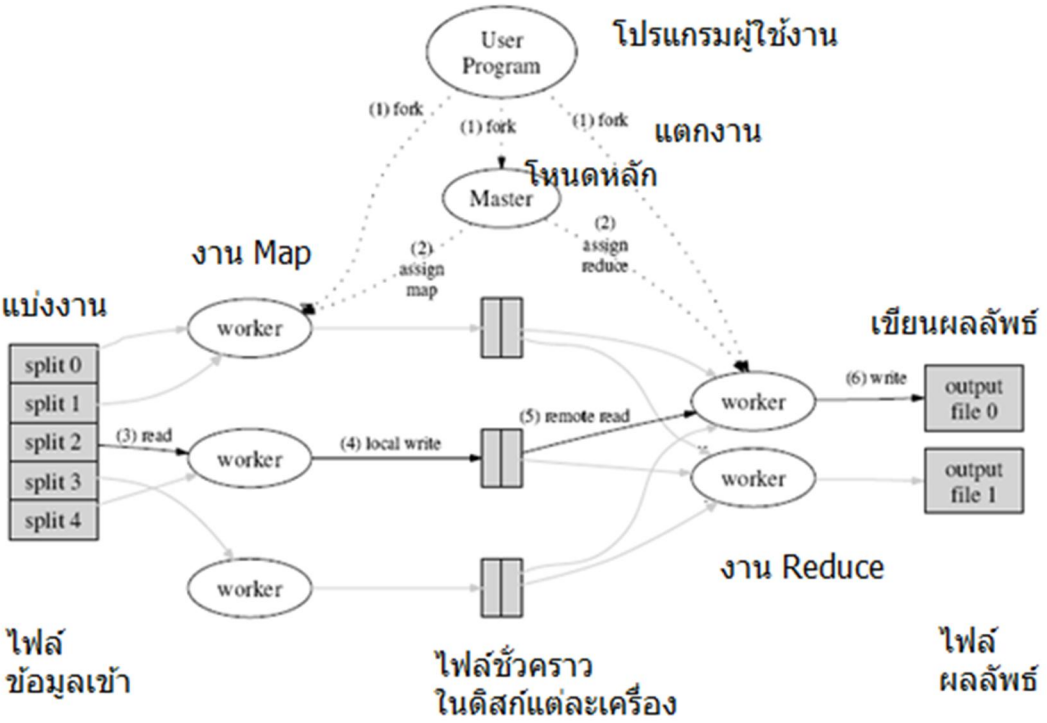
ตัว Hadoop ประกอบไปด้วยองค์ประกอบ 2 ส่วนหลักๆ ดังนี้

- **Hadoop Distributed File System (HDFS)** ซึ่งมีความสามารถในการจัดเก็บข้อมูลขนาดใหญ่โดยมีหลักการทำงานคือ แบ่งข้อมูลขนาดใหญ่ที่ต้องการจะเก็บออกเป็นส่วนย่อยๆ แล้วกระจายข้อมูลส่วนย่อยๆ นั้นไปยังเครื่องคอมพิวเตอร์ต่างๆ ที่เชื่อมต่อสำหรับการแก้ปัญหาในส่วนของการค้นหาข้อมูลภายในเทคโนโลยี HDFS วิธีที่เหมาะสมที่สุดเรียกว่าวิธี MapReduce
- **MapReduce Framework** ซึ่งเป็นส่วนของการปฏิบัติงานในการประมวลผลข้อมูลขนาดใหญ่ ซึ่งเป็นการส่งคำสั่งค้นหาแบบกระจายไปยังโหนดต่างๆ ในระบบ โดยที่ไม่จำเป็นต้องมีการย้ายข้อมูลระหว่างการประมวลผล ซึ่งจะทำให้การค้นหาข้อมูลมีประสิทธิภาพทางด้านความเร็วในการค้นหามากยิ่งขึ้น

MapReduce เป็นเฟรมเวิร์กในการเขียนโปรแกรมแบบหนึ่งที่ช่วยในงานประมวลผลที่มีชุดข้อมูลจำนวนมาก หลักการทำงานจะเป็นการทำงานแบบขนานซึ่งจะอาศัยหลายๆ โหนดช่วยกันทำงาน โดยที่ผู้ใช้งานนั้นไม่ต้องสนใจเบื้องหลังการทำงานเช่น วิธีการทำงานแบบขนาน วิธีการแบ่งข้อมูล การจัดการสมดุลของงาน การจัดการความคงทน เป็นต้น ในการทำงานแล้วผู้ใช้งาน MapReduce จะสนใจแค่ส่วนของ Map และส่วนของ Reduce ซึ่ง Map จะทำการจับคู่ของ Key และ Value ที่ต้องการ แล้วก็จะส่งไปให้ Reduce ทำการประมวลผลเพื่อให้ได้ผลลัพธ์รวมที่ต้องการ

MapReduce และ Google’s File System ได้คิดการประมวลผลที่เน้นการประมวลกับระบบข้อมูลขนาดใหญ่ โดยเน้นที่การค้นหาข้อมูลลักษณะเดียวกับ Search Engine โดยข้อมูลจะมีรูปแบบเดียวกันทำงานกับหลายๆ ตัว Mapper แต่เพื่อเอื้ออำนวยกับการโปรแกรมแบบขนานกรณีอื่นๆ อาจจะมีการเพิ่มการสร้างฟังก์ชัน Merge ขึ้นมา เพื่อรวมข้อมูลต่างๆ กันได้

การทำงานของ MapReduce คือจะกระจายงานต่างๆ ไปให้งาน Map อยู่บนแต่ละเครื่องทำงาน ซึ่งผู้ที่ควบคุมการกระจายงานก็คือโหนดหลัก Master โดยหลังจากที่ worker ทำงานเสร็จแล้วก็จะแจ้งโหนดหลัก เพื่อที่โหนดหลักจะส่งต่อผลการทำงานที่ได้จากการ Map ให้กับ Reduce worker เพื่อทำงานให้ได้ผลลัพธ์ต่อไป สามารถแสดงแผนผังการทำงานของ MapReduce ได้ดังภาพ 2.6

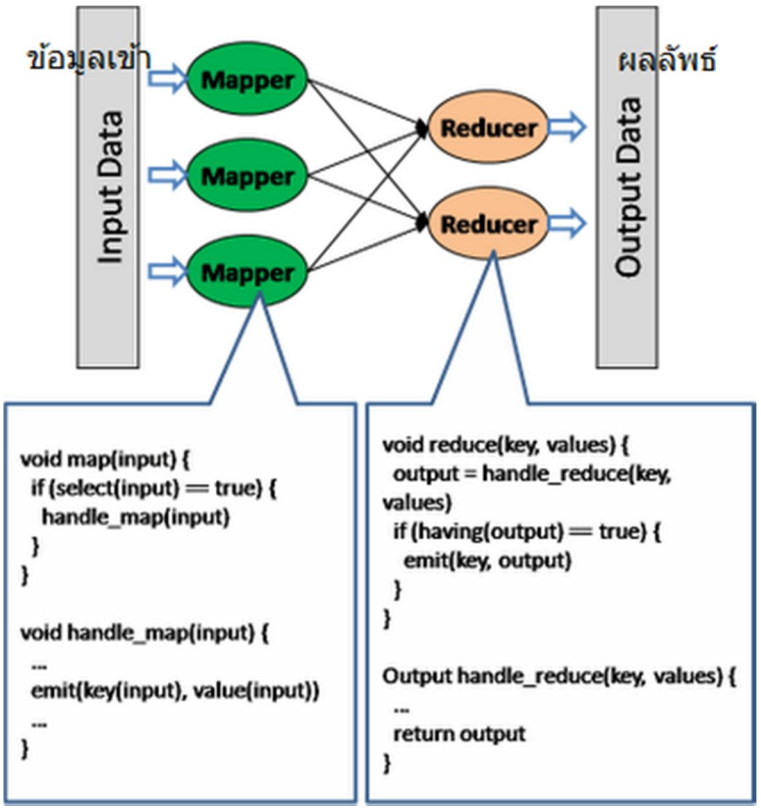


ภาพที่ 2.6 แผนผังการทำงานของ MapReduce [21]

จากภาพที่ 2.6 สามารถอธิบายการทำงานของ MapReduce ได้ดังต่อไปนี้

- เริ่มจากการ Map จะแบ่งแฟ้มข้อมูลที่เป็นข้อมูลเข้าเป็นส่วนย่อยๆ ขนาดประมาณ 16 MB ถึง 64 MB
- โหนดหลักหรือ Master จะพิจารณาว่ามี worker ตัวไหนว่างอยู่บ้าง ก็จะมอบงาน (assign task) ให้ ซึ่งก็จะมีทั้งงาน Map และ Reduce
- สำหรับ worker ที่ถูกมอบงาน Map ก็จะอ่านข้อมูลจากอินพุตแฟ้มข้อมูลที่ถูกแบ่งแล้ว Map Key และ Value ตามที่โปรแกรมผู้ใช้งานไว้
- ผลของการ Map จะถูกเก็บไว้ที่เครื่องที่ทำการ Map ซึ่งจะมีการแบ่งเป็นส่วนๆ ตาม Key ที่กำหนด และจะมีการส่งข้อความไปยัง Master เพื่อให้ Master ส่งไปทำงาน Reduce ต่อไป
- เมื่อ Reduce worker ได้รับแจ้งจาก Master จะทำการอ่านแฟ้มข้อมูลจากตำแหน่งที่ Master กำหนดให้มาแล้วทำการจัดเรียงข้อมูลตามลำดับ Key
- เมื่อเรียงข้อมูลตาม Key เสร็จแล้วก็จะทำตามฟังก์ชัน Reduce ตามที่ผู้ใช้งานกำหนดไว้ โดยจำนวนผลลัพธ์ที่ออกมาจะขึ้นอยู่กับจำนวนของ Reduce worker แล้วสุดท้ายก็จะมีการรวมแฟ้มผลลัพธ์เป็นแฟ้มเดียว
- เมื่อ MapReduce เสร็จแล้วก็จะแจ้งให้โหนด Master รู้ เพื่อแจ้งต่อไปยังโปรแกรมว่าทำ MapReduce เสร็จแล้ว

โมเดลการเขียนโปรแกรมสามารถแสดงได้ดังภาพที่ 2.7

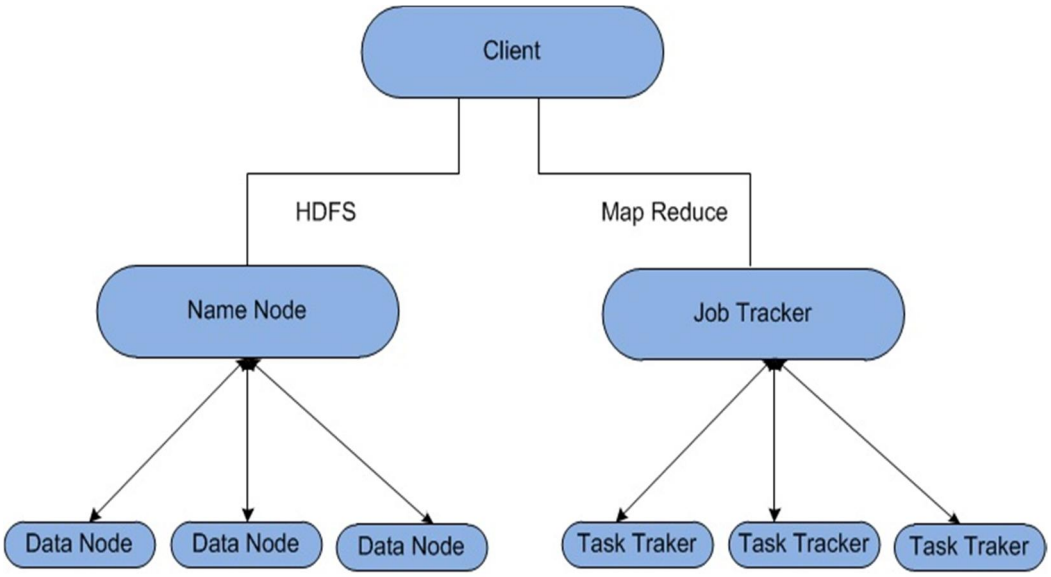


ภาพที่ 2.7 โมเดลการเขียนโปรแกรม [27,28]

เริ่มต้นด้วยการแบ่งงานที่รับเข้ามาเป็นส่วนๆ และทำการส่งแต่ละส่วนของงานนั้นไปยังเครื่องต่างๆ ซึ่งแต่ละเครื่องจะทำงานตาม Map script บนส่วนของข้อมูลที่บันทึกอยู่ โดย Map script จะป้อนข้อมูลบางส่วนและ Map ไปยังคู่ <key,value> ตามข้อกำหนด ตัวอย่างเช่น ถ้าต้องการนับความถี่ของคำในข้อความหนึ่ง key และ value อาจหมายถึง <word, count> เป็นต้น จากนั้น Map script จะเก็บคู่ของ <word,1> สำหรับ word ไว้ในที่เก็บอินพุตชั่วคราว (input stream) จุดประสงค์ของ Map script คือตัวแบบข้อมูลที่กำหนดให้ Reduce ทำการรวม จากนั้นคู่ <key,value> จะถูกสลับ (shuffled) โดยทั่วไปหมายถึงการที่คู่กับ key เดียวกันถูกจัดกลุ่มและส่งผ่านไปยังเครื่องเดียว ซึ่งจะทำงานในส่วนของ Reduce script ต่อไป นั่นคือ Reduce script จะเก็บรวบรวมคู่ <key,value> และทำการ Reduce ตาม Reduce script ที่ผู้ใช้ระบุ ในตัวอย่างของการนับคำนี้ต้องการนับจำนวนของคำที่เกิดขึ้นเพื่อให้สามารถได้มาซึ่งความถี่ ดังนั้นจึงทำการ Reduce script เพียงค่าผลรวมค่า 1 ในส่วนของ value ภายใน <key,value> ที่มี key เดียวกัน

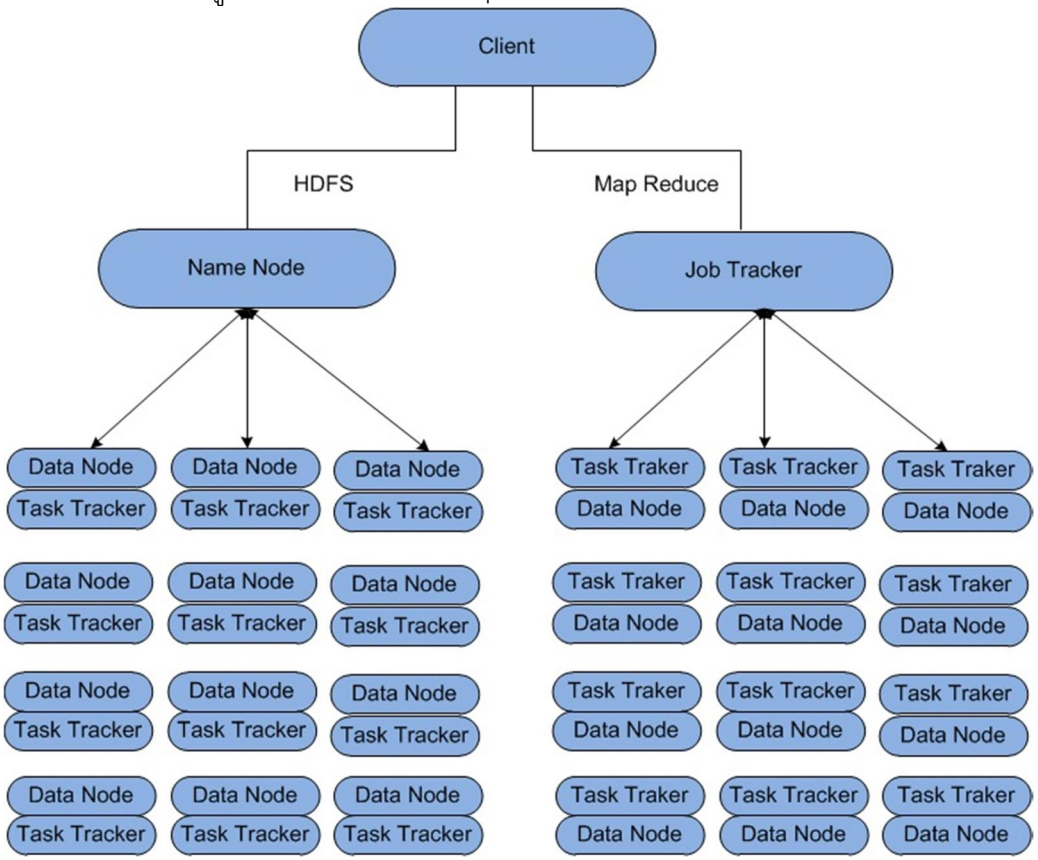
จากภาพที่ 2.8 ใน Hadoop จะสร้างจากโหนดหลักต่างๆ ดังนี้

- Name Node (Master)
- Data Node (Slave)
- Job Tracker
- Task Tracker



ภาพที่ 2.8 สถาปัตยกรรมระดับ Logical ของ Hadoop [29]

จากภาพที่ 2.8 เป็นสถาปัตยกรรมเชิง Logical ของ Hadoop แต่โดยกายภาพแล้ว Data Node และ Task Tracker สามารถถูกกำหนดบนเครื่องต่างๆ กันได้ ดังแสดงในภาพ 2.9



ภาพที่ 2.9 การวาง Data node และ Task Tracker บนเครื่องๆ เดียว [30]

การทำงานของ MapReduce ตาม Job Tracker และ Task Tracker

- ทำการรับงาน MapReduce ที่ถูกกำหนดโดยผู้ใช้
- การ Map งาน และ Reduce งานไปยัง Task Tracker
- ตรวจดูสถานะของงาน และ Task Tracker และเริ่มทำงานใหม่อีกครั้งหากงานนั้นไม่สำเร็จ

ในส่วน Task Tracker โหนดลูกของ MapReduce ทำงานดังนี้

- การ Map และ Reduce งานภายในคำสั่งที่ได้จาก Job Tracker
- จัดการพื้นที่และการส่งผลลัพธ์ชั่วคราว

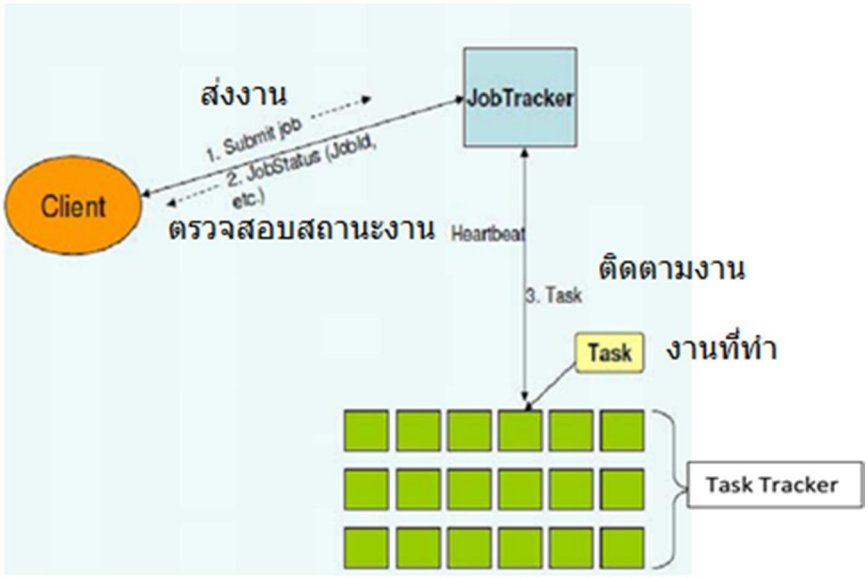
สำหรับ Job Tracker มีหน้าที่การทำงานได้แก่

- ดูแลงานทั้งหมด
- ทำการวางแผนตารางเวลาการทำงานทั้งหมด
- แบ่งงานออกเป็นงานย่อยๆ แล้วทำตามลำดับ
- จัดตารางงานบนโหนดที่ข้อมูลใกล้เคียงกัน โดยการแบ่งงานที่รับเข้ามา
- ตรวจดูแต่ละงาน
- ยกเลิกและเริ่มทำงานนั้นใหม่ในกรณีที่งานนั้นล้มเหลวหรือสูญหายไป

สำหรับ Task Tracker มีหน้าที่ดังต่อไปนี้

- ร้องของานใหม่
- ทำงาน
- ตรวจสอบสถานะของงาน
- รายงานผลการทำงานกลับโหนดหลัก

ซึ่งสามารถแสดงลำดับการทำงานได้ดังภาพ 2.10



ภาพที่ 2.10 ลำดับการทำงานของงาน MapReduce ใน Hadoop [30]

2.8 Java Concurrency [32]

การทำงานแบบพร้อมกัน (Concurrency) หมายถึง การทำงานหลายๆ งานในเวลาเดียวกัน สำหรับทั้งนี้ Java ได้ออกแบบมาเพื่อรองรับการทำงานหลายๆ งานในเวลาเดียวกัน ซึ่งทั้งหมดนี้จะอยู่ใน Package ของ java.util.concurrent

ในการพัฒนาโปรแกรมให้ทำงานแบบพร้อมกัน (Concurrent programming) ต้องเข้าใจพื้นฐานของโพรเซส และเทรด โดยสามารถอธิบายได้ดังนี้

- **โพรเซส** คือกระบวนการทำงานที่มีหน่วยความจำของตัวเอง หรือสามารถทำงานด้วยตัวเองได้เลย เช่น แอพพลิเคชันทั่วๆ ไปนั่นเอง หรือ อาจจะเป็นโพรเซสที่ไม่มีหน้าจอกราฟฟิกส์ หรือ User interface แต่อาจจะทำงานด้านหลังตลอดเวลา
- **เทรด** คือโพรเซสขนาดเล็ก (lightweight process) ซึ่งการใช้เทรด จะใช้ทรัพยากรของเครื่องคอมพิวเตอร์น้อยกว่าในการรันโพรเซส โดยทุกแอพพลิเคชันจะมีอย่างน้อยหนึ่งเทรดเสมอ ซึ่งอาจเรียกได้ว่าเทรดหลัก (main thread)

ปัญหาสำคัญที่เกิดจากการทำงานแบบพร้อมกันของหลายๆ เทรด ได้แก่

- การแทรกแซงระหว่างเทรด ตัวอย่างเช่นการเขียนค่าที่ผิดพลาดระหว่างสองเทรด สมมติมีตัวแปร a และมี 2 เทรดมาทำการเขียนค่าพร้อมกัน ค่าที่ได้จะเปลี่ยนไป นั่นคือ เริ่มจากค่า a มีค่าเป็น 0 จากนั้นถูกเปลี่ยนเป็น 1 โดย Thread1 ต่อมา Thread2 มาทำการลบออกไปทำให้ค่ากลับเป็น 0 อีก เป็นต้น

- ความสอดคล้องของค่าในหน่วยความจำ เป็นการดึงข้อมูลมาอ่านผิดพลาด เช่นจากตัวอย่างข้างต้น Thread1 ต้องการดึงข้อมูลมาดูเป็น 1 แต่ว่า Thread2 ได้ลบเป็น 0 แล้ว เป็นต้น

เทรดป็นลำดับการประมวลผลคำสั่งที่มีอยู่ภายในโพรเซส เนื่องจากเทรดนั้นมีคุณสมบัติบางส่วนของโพรเซสอยู่ด้วย จึงอาจเรียกได้ว่าเป็นโพรเซสขนาดเล็ก ในโพรเซสหนึ่งอาจประกอบด้วยเทรดจำนวนมากที่กำลังประมวลผลอยู่ โดยทั่วไปเทรดเป็นวิธีการที่นิยมนำมาใช้ในการเพิ่มประสิทธิภาพของการประมวลผลโพรเซสผ่านกลไกการประมวลผลแบบขนาน โดยที่หน่วยประมวลผลกลางจะสามารถสลับการทำงานระหว่างเทรดต่างๆ ได้อย่างรวดเร็ว ทำให้เกิดภาพเสมือนคล้ายกับว่าเทรดทั้งหมดนั้นกำลังประมวลผลอยู่ในเวลาเดียวกัน โพรเซสแบบเก่าจะเป็นโพรเซสที่ประกอบด้วยเทรดเพียงเทรดเดียว ซึ่งอาจจะอยู่ในสถานะใดๆ ก็เป็นได้ (running,blocked,ready หรือ terminated) เทรดแต่ละตัวจะมีสแตก (stack) เป็นของตนเองเนื่องจากเทรดมักจะเรียกใช้โพรซีเยอร์หรือฟังก์ชันต่างๆ กันจึงทำให้มีรายการทำงานที่แตกต่างกัน ระบบปฏิบัติการที่สนับสนุนการทำงานของเทรดจะกำหนดให้การประมวลผลในระดับพื้นฐานของหน่วยประมวลผลกลาง นั้นคือเทรดด้วยเนื่องจากเทรดไม่มีความเป็นอิสระจากเทรดอื่น (ภายในโพรเซสเดียวกัน) ดังนั้นเทรดในกลุ่มเดียวกัน (ภายในโพรเซสเดียวกัน) จึงใช้งานโค้ดและข้อมูล และทรัพยากรที่จัดสรรให้โดยระบบปฏิบัติการ เช่น การใช้งานแฟ้มข้อมูลต่างๆ ร่วมกัน

โปรแกรมภาษาจาวา (Java) จะทำงานภายในโพรเซสของตนเองภายใต้ Java Virtual Machine ซึ่งสนับสนุนการใช้งานได้หลายเทรดในหนึ่งโปรแกรมและถือเป็นส่วนหนึ่งของภาษาด้วย นอกจากนี้ Java ได้มีการจัดเตรียมการล๊อกโค้ดบางส่วนเพื่อป้องกันหลายๆ เทรดทำงานโค้ดส่วนเดียวกัน ทั้งนี้ใช้คำสำคัญ Synchronized ป้องกันส่วนของเมธอร์ดหรือทั้งเมธอร์ด การใช้วิธีการนี้ เรียกว่าเป็นการ Synchronization เป็นวิธีการหนึ่งของการสื่อสารระหว่างเทรด เพื่อสร้างลำดับการเข้าถึงทรัพยากรที่เหมาะสมที่ให้ผลลัพธ์การทำงานที่ถูกต้อง

ปัญหาที่ควรระวังในการใช้ Synchronization ก็คือ Deadlock นั่นคือเป็นการรอที่ไม่มีที่สิ้นสุดนั่นเอง เนื่องจากเวลาการใช้ Synchronized นั้นให้เทรดทำงานทีละตัว เพราะฉะนั้น ถ้าเมธอร์ดนั้นไปเรียกอีกเมเธอร์ดหนึ่งที่ถูก Synchronized อยู่อาจจะทำให้เกิด Deadlock ได้

ที่กล่าวมาข้างต้นเป็นการทำงานในแอพพลิเคชันขนาดเล็กเท่านั้น ในการทำงานแอพพลิเคชันขนาดใหญ่จะมีการใช้ Executor ซึ่งเป็น Java Interface ที่เตรียมไว้สำหรับทำหน้าที่ไปเรียกวัตถุให้ทำงาน มีความสามารถพิเศษในการจัดการกับวัตถุ Runnable จำนวนมากๆ จึงนิยมไว้ใช้กับโปรแกรมที่ต้องการรับมือกับเทรดจำนวนมาก และมีการสร้างกลุ่มของเทรด (Thread Pool) ก็เป็นการกำหนดจำนวนของเทรดที่ใช้ทำงานโดยสามารถกำหนดเทรดที่เกี่ยวข้องเท่านั้นให้ทำงาน

<u>เฟรมเวิร์ก Fork-Join ใน Java 7</u>

ใน Java เวอร์ชัน 7 ได้แนะนำกลไกการทำงานแบบขนานแบบใหม่เพื่อใช้ในการคำนวณงานที่เข้มข้นขึ้น นั่นคือเฟรมเวิร์ก Fork-Join โดยช่วยให้สามารถแจกจ่ายงานบางงานไปยังหลาย worker และคอยผลลัพธ์โดย Fork-Join เป็นโครงสร้างที่สนับสนุนการเขียนโปรแกรมแบบขนาน ซึ่งเป็นเทคนิคที่ออกแบบมาง่ายและมีประสิทธิภาพมากที่สุดเพื่อให้ได้การทำงานแบบขนานที่มีประสิทธิภาพดี โดยอัลกอริธึมของ Fork-Join เป็น

อัลกอริธึมแบบขนานที่อยู่ในกลุ่มของอัลกอริธึมแบบแบ่งแยกและเอาชนะ (divide-and-conquer) สามารถแสดงได้ตามโค้ดดังต่อไปนี้

```
Result solve(Problem problem) {
        if (problem is small)
                directly solve problem
        else {
                split problem into independent parts
                fork new subtasks to solve each part
                join all subtasks
                compose result from subresults
        }
}
```

จากโค้ดข้างต้นขั้นตอน Fork เริ่มทำงานโดยทำการแบ่งงานออกเป็นงานย่อยๆ ส่วน Join จะเกิดขึ้นเมื่องานปัจจุบันไม่สามารถดำเนินการได้จนกระทั่งงานย่อยๆ ที่ถูก Fork สำเร็จ โดยวิธีการ Fork-Join จะทำงานคล้ายกับการแบ่งแยกและเอาชนะและเป็นการเรียกตัวเองซ้ำ (Recursive) แบ่งงานออกเป็นงานย่อยๆ วนซ้ำไปเรื่อยๆจนกระทั่งปัญหานั้นเล็กเพียงพอที่จะสามารถแก้ได้ง่ายโดยวิธีการแบบลำดับสั้นๆ

เฟรมเวิร์ก Fork/Join เป็นรูปแบบใหม่ที่ถูกเพิ่มในแพลตฟอร์ม Java 7 และเป็นหนึ่งในข้อดีหลักใน Java 7 โดยในเฟรมเวิร์ก Fork/Join มีการเพิ่มคลาส 2 คลาสหลักๆเข้าไปใน Package ของ java.util.concurrent นั่นคือ

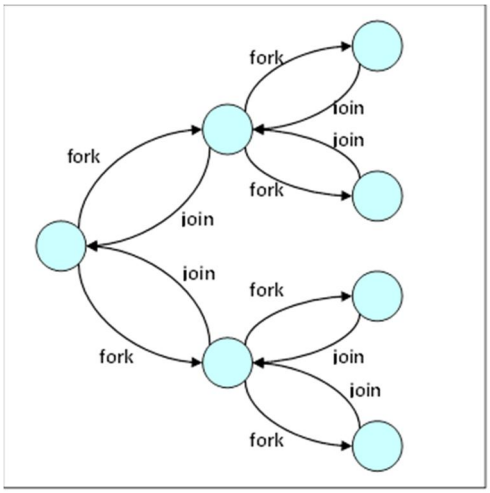
- ForkJointPool
- ForkJoinTask

คลาส ForkJoinPool เป็นคลาส Executor และรันอินสแตนท์ของ ForkJoinTask หลักการทำงานหลักๆ ของ ForkJointPool คือทำงานแบบ “work stealing” ซึ่งเทรดจะพยายามค้นหาและดำเนินการกับงานย่อยซึ่งถูกสร้างโดยงานอื่นที่ยังคงทำงานอยู่ โดยอินสแตนท์ของ ForkJoinTask เป็นเทรดขนาดเล็กมากเมื่อเทียบกับเทรด Java ปกติ เมื่อ ForkJoinTask จะเริ่มต้นทำงาน จะเริ่มที่งานย่อย แล้วรอให้งานย่อยที่กำลังทำงานอยู่เสร็จสิ้น ซึ่งคลาส Executor ForkJoinPool เป็นผู้รับผิดชอบสำหรับการกำหนดงานย่อยนี้ไปยังอีกเทรดหนึ่งภายใน ThreadPool ซึ่งขณะนี้จะคอยให้บางงานทำงานเสร็จ โดยเทรดที่กำลังทำงานอยู่ใน ThreadPool จะพยายามทำงานย่อยอื่นที่ถูกสร้างโดยงานอื่นๆ ซึ่งจะขึ้นอยู่กับจำนวนของโพรเซสเซอร์ที่สามารถหามาได้ ForkJoinPool พยายามที่จะทำงานร่วมกันในระดับการทำงานแบบขนาน สำหรับ ForkJoinTask มีฟังก์ชันหลักหรือเมธอร์ดที่เพิ่มเติมไป คือ

-fork() เป็นเมธอร์ดที่ตัดสินใจบนการทำงานแบบ Asynchronous ของ ForkJoinTask โดย เมธอร์ดนี้สามารถสร้างงานใหม่ได้

-join() เป็นเมธอร์ดที่รับผิดชอบในการส่งคืนค่าผลลัพธ์ที่ผ่านการคำนวณสำเร็จสมบูรณ์แล้ว โดยยอมให้งานหนึ่งคอยให้อีกงานหนึ่งทำงานสำเร็จก่อน

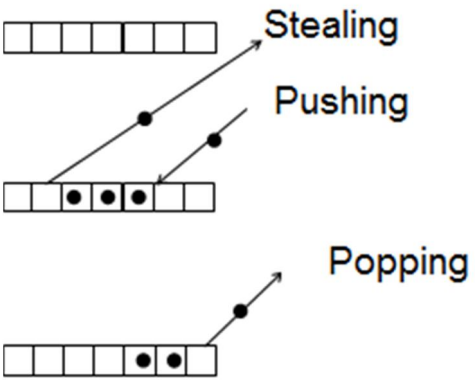
ซึ่งสามารถแสดงภาพการทำงานงานของ Fork/Join โดยรวมได้ดังภาพ 2.11



ภาพที่ 2.11 แสดงภาพรวมของการทำงานของ Fork/Join

(ที่มา - Java 7 Fork/Join Framework. Accessed December 25.
http://www.developer.com/java/java-7-forkjoin-framework.html)

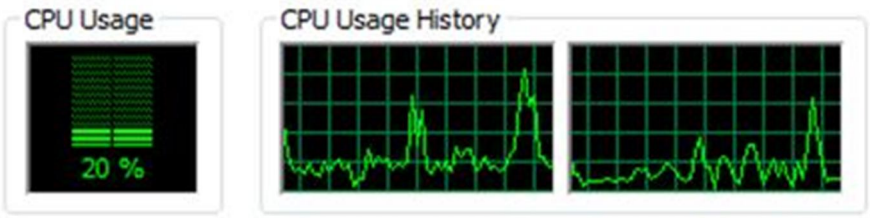
หัวใจสำคัญของ Fork/join อยู่ที่กลไกการจัดตารางงาน แต่ละเทรดทำงาน โดยมีการเก็บงานที่จะทำไว้ในคิวงานของตัวเอง งานย่อยถูกทำตามลำดับแบบ LIFO (Last In First Out) ถ้าในคิวนั้นไม่มีงานที่จะต้องทำมันจะชิงงานจากโหนดอื่นในแบบ FIFO (First In First Out) ดังภาพ 2.12 ที่เรียกว่า Work Stealing [32]



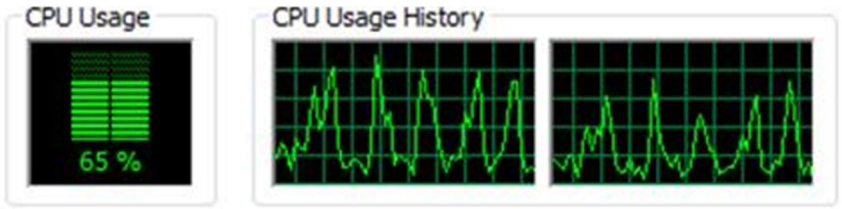
ภาพที่ 2.12 กระบวนการ Work Stealing

นอกจากนี้สิ่งที่สำคัญมากที่จะทราบว่าการใช้งานหน่วยประมวลผลกลางในกรณีของ Single thread จะมีประสิทธิภาพน้อยกว่าเมื่อเปรียบเทียบกับการใช้เฟรมเวิร์ก Fork/Join ซึ่งสามารถแสดงได้ดังภาพ 2.13

เทรดเดียว (Single Threaded Model)



เฟรมเวิร์ก Fork/Join



ภาพที่ 2.13 การทำงานของหน่วยประมวลผลกลางระหว่างเทรดเดียว (Single Threaded Model) กับเฟรมเวิร์ก Fork/Join

(ที่มา - http://www.developer.com/java/java-7-forkjoin-framework.html)

การเขียนโปรแกรมแบบขนานในภาษา Java พัฒนามาจากตัวแบบเทรดเดียวมาเป็นเฟรมเวิร์ก Fork/Join ที่ถูกแนะนำใน Java 7 โดยตัว Executor ที่เป็นการปรับปรุงที่สำคัญที่เกิดขึ้นกับ Java ตั้งแต่ Java 5 แต่ไม่สามารถถูกใช้สำหรับการคำนวณแบบขนานได้ ดังนั้นเฟรมเวิร์ก Fork/Join ใน Java 7 จึงเป็นพัฒนาการที่สำคัญ ซึ่งในงานวิจัยนี้ได้ใช้ Java Concurrency Package ในการพัฒนาส่วนการของการดึงข้อมูลแบบขนานด้วย

บทที่ 3 การดำเนินการวิจัย

เป้าประสงค์ของการทำงานโครงการย่อยที่ 1 ประกอบด้วย การสำรวจซอฟต์แวร์ที่ใช้ในปัจจุบันของหน่วยงาน แหล่งที่มาของข้อมูลของการท่องเที่ยวเชิงสุขภาพ ขั้นตอนหรือกระบวนการรวบรวมข้อมูล การศึกษางานวิจัยที่เกี่ยวข้องด้านการประมวลแบบขนานบนกลุ่มเมฆ พัฒนาอัลกอริธึมแบบขนาน และการแทนค่าข้อมูลสำหรับการประมวลผล การออกแบบหน้าจอ การติดต่อผู้ใช้ การเชื่อมต่อกับข้อมูล การประเมินผล ผลการค้นหา ด้านประสิทธิภาพ รวมทั้งประเมินผลการใช้งานกับผู้ใช้ในบทบาทต่างๆ กัน สรุปเป็นขั้นตอนการดำเนินการดังนี้

1. การสำรวจลักษณะซอฟต์แวร์ ข้อมูลที่ใช้อยู่ปัจจุบันของเทศบาล

คณะผู้วิจัยได้ สอบถามข้อมูลซอฟต์แวร์ที่ใช้อยู่ในเทศบาล และการทำงานของหน่วยงาน แหล่งข้อมูลต่างๆ ของการท่องเที่ยวที่ทางเทศบาลใช้อยู่ ด้วยการสัมภาษณ์และแบบสอบถาม โดยรายละเอียดอยู่ในรายงานแผนการวิจัย หัวข้อ 4.1 และ 4.2 และในภาคผนวกของแผนการวิจัย

2. การศึกษางานวิจัยและทฤษฎีที่เกี่ยวข้องของการประมวลผลบนกลุ่มเมฆ และการทำงานแบบขนาน

คณะผู้วิจัยได้ศึกษารายละเอียดต่างๆ ดังในบทที่ 2 และ บทที่ 3

3. การพัฒนาอัลกอริธึมแบบขนาน การแทนค่าข้อมูลในการประมวลผล

คณะผู้วิจัยได้ทำการพัฒนาอัลกอริธึมแบบขนานในส่วนของการประมวลผลข้อมูลเข้า และส่วนของการค้นหา

4. การออกแบบหน้าจอต่อผู้ใช้งาน การเชื่อมต่อข้อมูลต่างๆ

คณะผู้วิจัยได้ทำการออกแบบหน้าจอ การเชื่อมต่อกับฐานข้อมูล เซอร์วิสจากภายนอก รายละเอียดหน้าจออยู่ในภาคผนวก ซ. ของแผนงาน คู่มือผู้ใช้งาน

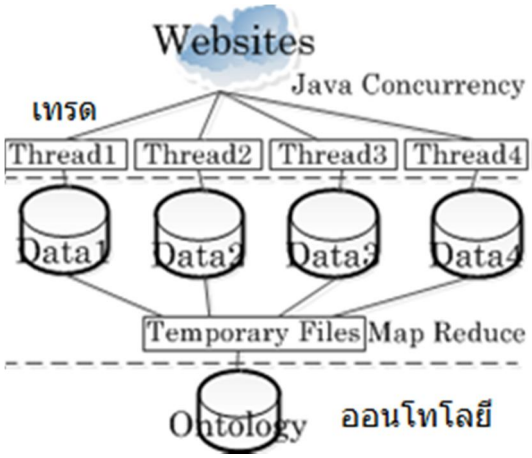
5. การประเมินสิทธิภาพ

คณะผู้วิจัยได้ทำการประเมินประสิทธิภาพด้านเวลาใน 2 แบบ ได้แก่ เวลาที่ใช้ในการประมวลผลคิวรี่ และเวลาที่ใช้ในการประมวลผลแบบขนานเมื่อนำข้อมูลตัวอย่างเข้าไปในสถาปัตยกรรมแบบขนาน เช่น หน่วยประมวลผลกราฟิก และทดสอบเวลาการค้นหา

6. การประเมินผู้ใช้งาน

คณะผู้วิจัยได้ทำการประเมินผลจากผู้ใช้งานความเห็นต่อหน้อจอแสดงผลเมื่อระบบพัฒนาเสร็จสิ้นความเห็นต่อการนำเสนอข้อมูลและความเห็นต่อการอบรมผู้ใช้งานในส่วนของเทศบาลดังในภาคผนวก ฌ. ของแผนงานวิจัย

3.1 สถาปัตยกรรมโครงสร้างสำหรับการนำข้อมูลเข้าแบบขนาน



ภาพที่ 3.1 สถาปัตยกรรมโครงสร้างการนำข้อมูลเข้าแบบขนาน

สถาปัตยกรรมโครงสร้างการนำข้อมูลเข้าแบบขนานประกอบไปด้วย 3 ส่วน

ส่วนที่ 1: การใช้ Java Concurrency ทำงานแบบหลายเทรดพร้อมกัน เพื่อไปทำการค้นหาข้อมูลจากเว็บไซต์ที่กำหนดภายในเวลาพร้อมกัน จำนวนเทรดถูกกำหนดไว้ในโปรแกรม ข้อมูลที่ได้จากแต่ละเทรดถูกเก็บแยกกัน เพื่อนำไปประมวลผลในขั้นตอนต่อไป

ส่วนที่ 2: การใช้ MapReduce ทำการประมวลผลข้อมูลของแต่ละเทรด และรวบรวมผลลัพธ์ไว้ในแฟ้มข้อมูล

ส่วนที 3: การใช้ออนโทโลยีจัดเก็บข้อมูลที่ได้นำเข้ามา

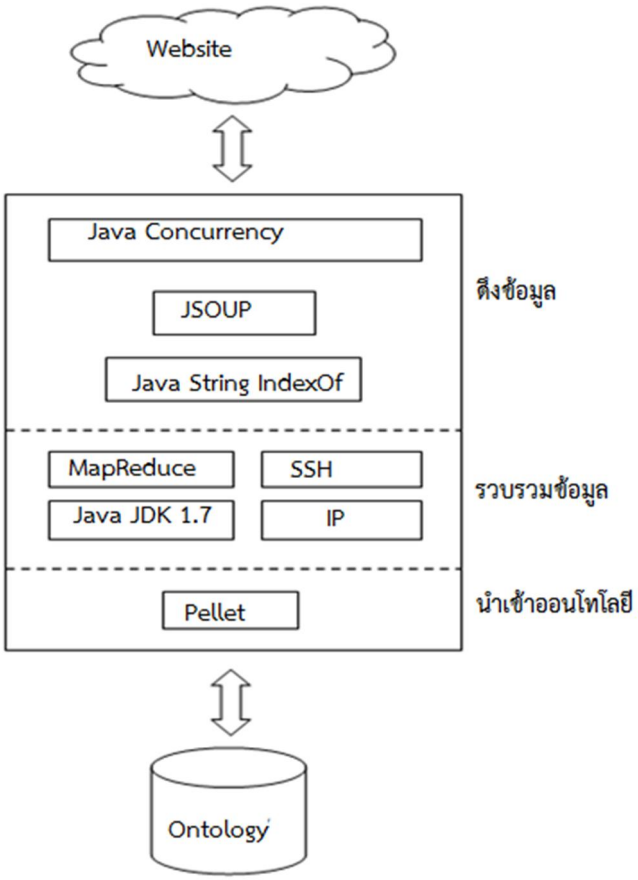
การดึงข้อมูลเข้าจากเว็บไซต์เพื่อจัดลงออนโทโลยีเป็นการทำงานโดยผู้ดูแลระบบ โดยทำการดึงข้อมูลจากเว็บไซต์ เลือกข้อมูลส่วนที่ต้องการ จากนั้นจึงรวมข้อมูลเข้าด้วยกันเพื่อจัดรูปแบบข้อมูล และบันทึกข้อมูลลงตามโครงสร้างออนโทโลยีที่สร้างไว้

ข้อมูลที่ได้รับจากแต่ละเว็บไซต์มีดังในภาคผนวก ข. ของแผนงานวิจัยต่อไปนี้

- ข้อมูลจาก Agoda คือ โรงแรม ที่อยู่ จำนวนห้อง และพิกัดทางภูมิศาสตร์
- ข้อมูลจาก AtSiam คือ โรงแรม ราคา หมายเลขโทรศัพท์ พิกัดภูมิศาสตร์และสิ่งอำนวยความสะดวก
- ข้อมูลจาก Hey Huahin คือ ชื่อร้านหมวดกลุ่มสปาและนวดแผนไทย ที่อยู่และพิกัดทางภูมิศาสตร์
- ข้อมูลจาก TripAdvisor คือโรงแรม ที่อยู่ ราคา ระดับดาว และสิ่งอำนวยความสะดวก

ตัวอย่างสิ่งอำนวยความสะดวก อาทิเช่น สปาและนวดแผนไทย ที่จอดรถ ไวไฟ บริการตามห้อง บาร์ ที่จอดรถ ซักรีด สระว่ายน้ำ สถานที่ออกกำลังกาย ห้องประชุม ห้องพยาบาล สวน และสินค้าที่ระลึก เป็นต้น

สถาปัตยกรรมซอฟต์แวร์ของการดึงข้อมูลแสดงในภาพที่ 3.2



ภาพที่ 3.2 สถาปัตยกรรมซอฟต์แวร์

ภาพที่ 3.2 แสดงสถาปัตยกรรมซอฟต์แวร์ที่ใช้ในระบบ จะเห็นว่าสถาปัตยกรรมแบ่งออกเป็น 3 ระดับชั้น คือ การค้นหาและการเข้าถึงข้อมูล การรวบรวมข้อมูล และการนำข้อมูลเข้าออนโทโลยี

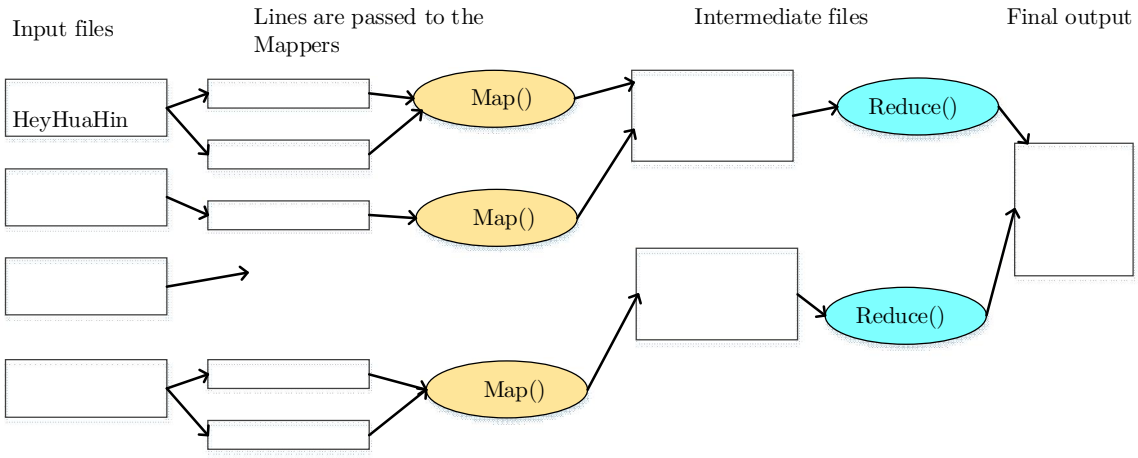
ระดับชั้นที่ 1: การค้นหาและการเข้าถึงข้อมูล

คณะผู้วิจัยใช้ Java Concurrency ช่วยในการเข้าถึงข้อมูลโรงแรมจากเว็บไซต์ที่กำหนด เพื่อช่วยให้ระบบสามารถทำงานด้วยความรวดเร็ว และใช้ Jsoup ช่วยในการดึงข้อความจากหน้าเว็บที่เป็นภาษา HTML จำนวน 3 แหล่งข้อมูล และบันทึกลงแฟ้มข้อมูล (text file) เนื่องจากแต่ละแหล่งข้อมูลอาจจะมีลิงค์ย่อย คณะผู้วิจัยใช้เทรด Java ในการดึงข้อมูลของแต่ละลิงค์ และนำข้อมูลที่ดึงจากแหล่งข้อมูลเดียวกันบันทึกไว้ในแฟ้มข้อมูลเดียวกัน เมื่อได้แฟ้มข้อมูล 3 แฟ้ม จากแต่ละแหล่งข้อมูลเรียบร้อยแล้ว คณะผู้วิจัยได้ใช้คำสั่ง

จาก Java String คำสั่ง indexOf Parsing ช่วยในการค้นหาและตัดเฉพาะข้อความที่ต้องการจากตัวเอกสารที่ดึงมาได้ทั้งหมด

การทำงานในสถาปัตยกรรมระดับชั้นนี้ จะได้ข้อมูลของโรงแรมที่ประกอบไปด้วย ชื่อภาษาไทย ชื่อภาษาอังกฤษ ที่อยู่ รหัสไปรษณีย์ ราคาเริ่มต้น จำนวนดาวโรงแรม จำนวนห้องพัก คะแนนความนิยม หมายเลขโทรศัพท์ ละติจูด ลองจิจูด และสิ่งอำนวยความสะดวกของโรงแรม โดยที่ระบบจะบันทึกสิ่งอำนวยความสะดวกก็ต่อเมื่อโรงแรมมีสิ่งอำนวยความสะดวกนั้นๆ แต่ถ้าหากโรงแรมไม่มีสิ่งอำนวยความสะดวกนั้นๆ ระบบจะบันทึกเป็นช่องว่าง และเพื่อให้ง่ายในการจัดการข้อมูลจำนวนมาก ระบบจะทำการบันทึกสิ่งอำนวยความสะดวกต่างๆ เรียงตามลำดับ ดังนี้ บริการรถรับ-ส่งสนามบิน บาร์/เลาจ์ บาร์บีคิว ชายหาด บริการรถโดยสาร ร้านกาแฟ บริการรับเลี้ยงเด็ก สวน อินเทอร์เน็ต บริการซักรีด มุมนั่งเล่น นวด ห้องประชุม มินิบาร์ ภูเขา ไนท์คลับ/คาราโอเกะ ที่จอดรถ ตู้เย็น ร้านอาหาร รูมเซอร์วิส ตู้เซฟ ทะเล บริการจัดเลี้ยง สปา ครัวขนาดเล็ก สระว่ายน้ำ โทรศัพท์ และอาหารเช้า ข้อมูลแต่ละส่วนถูกคั่นด้วยจุลภาค (,)

ข้อมูลที่บันทึกในแฟ้มข้อมูลทั้ง 3 แฟ้ม มีการเรียงลำดับข้อมูลเหมือนกันตามที่อธิบายไว้ข้างต้น หากไม่มีข้อมูลจะบันทึกเป็นช่องว่างแทน ทั้งนี้เพื่อความสะดวกในการจัดการข้อมูลจำนวนมาก คณะผู้วิจัยได้เพิ่มสัญลักษณ์และเครื่องหมายทวิภาค (:) ไว้ข้างหน้าข้อความเพื่อระบุว่าข้อความส่วนนี้เป็นอะไร เช่น room: 20 ให้ความหมายได้ว่ามีจำนวนห้อง 20 ห้อง



ภาพที่ 3.3 ขั้นตอน MapReduce

ระดับชั้นที่ 2: การรวบรวมข้อมูล

คณะผู้วิจัยใช้กระบวนการทำงานของ MapReduce ในการปรับรวบรวมข้อมูลที่ดึงจากแหล่งข้อมูล 3 แหล่ง ซึ่งอาจจะมีบางโรงแรมที่ได้ข้อมูลออกมาเหมือนกัน โดยที่ MapReduce แบ่งการทำงานเป็น 2 ส่วน คือ Map จะอ่านข้อมูลจากแฟ้มข้อมูลในขั้นแรก แล้วระบุให้ชื่อโรงแรมภาษาไทยเป็น key (ชื่อภาษาไทย คือ ข้อมูลส่วนแรกซึ่งได้ประโยชน์จากการแบ่งข้อมูลด้วยจุลภาค) เพื่อใช้ในการจำแนกว่ามีโรงแรมใดบ้างที่ใช้ชื่อเดียวกัน ในขั้นตอนนี้จะระบุได้เฉพาะชื่อที่เขียนเหมือนกันทุกตัวอักษรเท่านั้น เช่น คำว่า “รี

สอร์ท” กับ “รีสอร์ต” การ Map จะมองเป็นคนละชื่อกัน และเพื่อช่วยให้การทำงานง่ายขึ้น จึงมีการตัดการเว้นวรรคภายในชื่อโรงแรมออกไปด้วย เพราะหากมีการเว้นวรรคกับไม่มีการเว้นวรรค Map ก็จะมองเป็นชื่อที่ต่างกันอีกเช่นกัน ส่วนชื่อภาษาอังกฤษและข้อมูลโรงแรมส่วนที่เหลือจะถูกกำหนดให้เป็นค่า value ผลลัพธ์ของการ Map ระบุเป็นคู่ <key, value> และส่งออกไปเป็นอินพุตให้ Reduce จำนวน 1 แฟ้มเท่านั้น ไม่ว่าจะมีจำนวนแฟ้มข้อมูลเท่าไร การ Map จะรวบรวมข้อมูลออกมาได้เพียง 1 แฟ้ม

ในขั้นตอนการ Reduce จะนำค่า value ที่มี key เดียวกันมาต่อกัน คือ หากมีชื่อโรงแรมที่เขียนเหมือนกันทุกประการแล้ว ข้อมูลต่างๆ (ค่า value) จะถูกนำมาต่อกันเท่ากับที่หาได้ทั้งหมดในแฟ้มข้อมูล (ผลลัพธ์ของ Map) เมื่อต่อกันของ 1 แฟ้มแล้วจะจบด้วยเครื่องหมายอัฒภาค(;) ทั้งนี้หากไม่พบข้อมูลชื่อโรงแรมที่เขียนเหมือนกัน ก็จะมีแต่ข้อมูลจากหน้าเว็บเดียวที่ต่อท้ายด้วยเครื่องหมายอัฒภาค (สามารถรู้ได้ว่ามีข้อมูลจากแหล่งเดียว ไม่มีการต่อกัน) ในขั้นตอนนี้ไม่สามารถบอกได้ว่ามีลำดับการต่อกับของข้อมูลอย่างไร เป็นกระบวนการภายในของ Reduce เอง บางครั้งเราจะพบว่ามีการต่อกันของข้อมูลจาก Atsiam; Tripadvisor; Agoda; หรือ Atsiam; Agoda; Tripadvisor; หรือหากมีข้อมูลเพียงแหล่งเดียวก็จะเป็น Tripadvisor; หรือ Agoda; เพียงอย่างเดียว ผลลัพธ์ของ Reduce ถูกระบุเป็นคู่ <key, value> เช่นเดียวกับการ Map

การทำงานในขั้นตอนนี้ คณะผู้วิจัยได้ทำการลดจำนวนข้อมูลที่ซ้ำซ้อนกันได้ระดับหนึ่งเท่านั้น กล่าวคือ จะได้ข้อมูลของโรงแรมที่ไม่ซ้ำซ้อนกัน เฉพาะกรณีที่มีการเขียนชื่อโรงแรมเดียวกันเหมือนกันเท่านั้น ซึ่งเป็นข้อจำกัดจากการที่ได้ข้อมูลมาตั้งแต่แรกแล้วเพราะแต่ละหน้าเว็บมีการเขียนแตกต่างกัน

ต่อมาจึงทำการเติมข้อมูลที่ไม่มีจากข้อมูลที่มี กล่าวคือ ข้อมูลโรงแรมจากเว็บหนึ่งมีบ้าง ไม่มีบ้าง ทำให้ข้อมูลขาดหายไป คณะผู้วิจัยจึงทำการเติมข้อมูลส่วนที่ไม่มีด้วยข้อมูลที่มี ที่ได้จากการดึงข้อมูลจากหน้าเว็บอื่น เช่น จาก Agoda ไม่มีข้อมูลหมายเลขโทรศัพท์ ก็จะไปหาข้อมูลหมายเลขโทรศัพท์จากข้อมูลที่ต่อท้ายมาเติม (ผลลัพธ์ของ MapReduce) หากไม่มีก็จะถือว่าข้อมูลในส่วนนั้นไม่สามารถระบุได้จริงๆ ในการไปหาข้อมูลมาเติมนั้น ทำการเติมเข้าไปในข้อมูลชุดแรก (แต่ละชุดคั่นด้วยเครื่องหมายอัฒภาค) โดยทำเฉพาะส่วนที่ไม่มีอยู่แล้วตั้งแต่แรก (พิจารณาข้อความหลังเครื่องหมายจุลภาค) เมื่อพบเป็นช่องว่าง ก็จะรู้ได้ว่าเป็นส่วนที่ขาดจากสัญลักษณ์ใด แล้วจึงไปหาในข้อมูลที่ต่อท้ายที่มีชื่อสัญลักษณ์เดียวกัน หากพบครั้งแรกก็นำมาเติมทันที ไม่ต้องหาในส่วนถัดไป หากไม่พบก็จะค้นหาในลำดับถัดไป (หาเฉพาะที่เป็นชื่อสัญลักษณ์เดียวกัน) ผลลัพธ์จากขั้นตอนนี้จะได้ข้อมูลโรงแรมที่มากขึ้น และมีลำดับเช่นเดียวกับตอนที่นำเข้ามา แต่เหลือเพียงส่วนเดียว ไม่มีเครื่องหมายอัฒภาค ได้ผลลัพธ์ออกมา 1 แฟ้มข้อมูล

3.2 การพัฒนาอัลกอริธึมการประมวลผลคิวรี่แบบขนาน และการแทนค่าข้อมูล

งานวิจัยนี้ทำการประมวลผลคิวรี่แบบขนานโดยใช้หน่วยประมวลผลกราฟิกส์ที่มีการทำงานแบบพร้อมกัน และใช้หน่วยความจำร่วมกัน จำนวนหน่วยประมวลผลกราฟิกส์ที่ใช้ขึ้นกับลักษณะของการ์ดจอ (Graphic card) ที่มีอยู่ การนำข้อมูลเข้าไปให้หน่วยประมวลผลกราฟิกส์ช่วยคำนวณนั้นจะมีขั้นตอนการแปลงข้อมูลก่อนนำเข้า ประกอบด้วยขั้นตอนย่อยดังต่อไปนี้ (ภาพรวมเป็นดังภาพที่ 3.4 (ก))

- ข้อมูลนำเข้า คือ ออนโทโลยีด้านการท่องเที่ยวเชิงสุขภาพ

- ข้อมูลคำถาม คือ คิวรี่คำถามจากผู้ใช้งาน
- ข้อมูลผลลัพธ์ คือ คำตอบหลังจากผ่านการคิวรี่ ส่งจากระบบแก่ผู้ใช้งาน

ภาพที่ 3.4 (ก) แสดงให้เห็นการทำงานของระบบเว็บเชิงความหมายแบบเดิมที่สามารถคิวรี่ระหว่างโปรแกรมเว็บและ RDF ที่เป็นโครงสร้าง XML ผ่านมิดเดิลแวร์ดังภาพที่ 3.4 (ข) และแปลงเป็นเว็บแอพพลิเคชันสำหรับผู้ใช้ได้เลย

ภาพที่ 3.4 (ค) และ (ง) โปรแกรมเบื้องหลังที่ได้รับการปรับปรุง จะทำงานระหว่างหน่วยประมวลผลกลาง หน่วยประมวลผลกราฟิกส์และชุดข้อมูลเพื่อลดขนาดของข้อมูลให้สามารถนำเข้าไปประมวลผลด้วยหน่วยประมวลผลกราฟิกส์ได้ และเร่งความเร็วด้วยการค้นหาแบบขนานในการคิวรี่

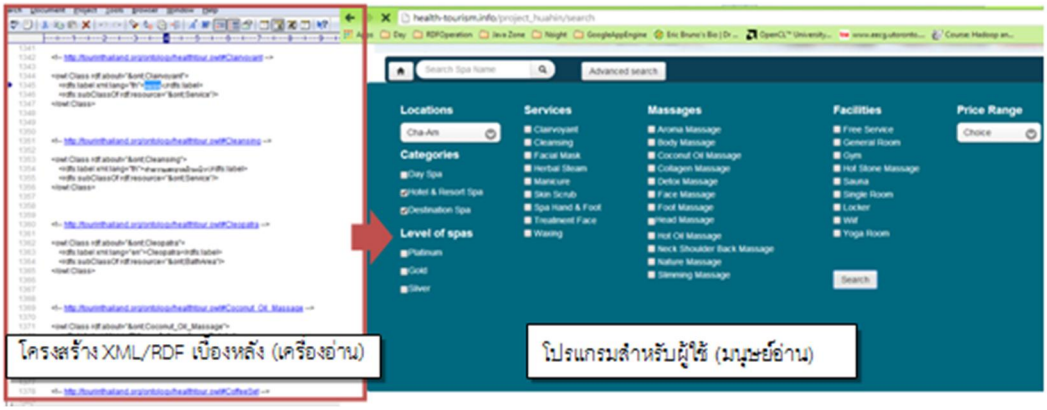
ในภาพที่ 3.4 (ง) มีรายละเอียดดังนี้

ขั้นตอนที่ 1 นำข้อมูลออนโทโลยีเข้าไปโดยแปลงเป็นในโครงสร้าง RDF และแปลงสู่รูปแบบย่อแบบไบนารีก่อน

ขั้นตอนที่ 2 นำส่วนเนื้อหามาแปลงเป็นเลขสำหรับดัชนีแบบแบ่งเป็นชุดๆ จากนั้นแปลงเป็นลักษณะดัชนี 1-0 เพื่อเข้าสู่หน่วยความจำของประมวลผลกราฟิกส์ เพื่อประมวลผล โดยนำข้อมูลคิวรี่ที่ต้องการค้นหาที่รับจากหน้าจอผู้ใช้มาแปลงเป็นลักษณะเดียวกันเพื่อค้นหาในหน่วยความจำดังกล่าว

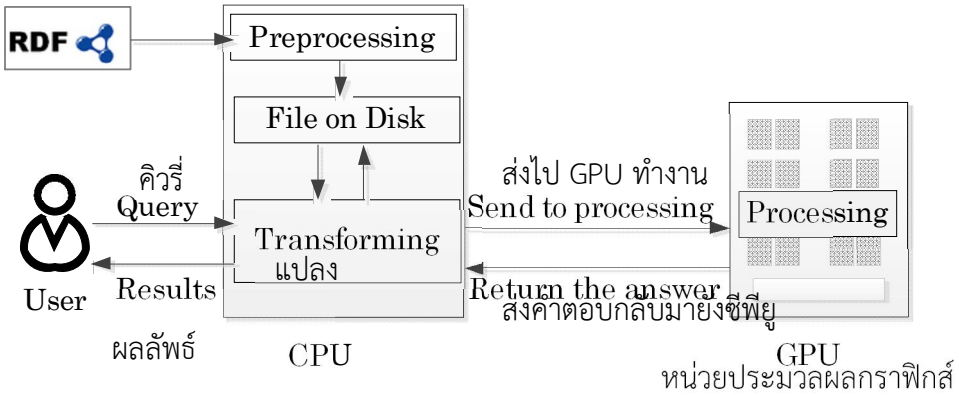
ขั้นตอนที่ 3 นำข้อมูลที่ค้นหาพบออกมาจากหน่วยความจำของหน่วยประมวลผลกราฟิก แล้วแปลงกลับด้วยสคริปต์ต่าง ๆ เพื่อให้ผู้ใช้ระบบสามารถอ่านได้ผ่านเว็บเบราว์เซอร์

(ก) ขั้นตอนโดยรวม



(ข) จากหน้าเวบด้านขวาจะสามารถแปลงเป็น RDF ด้านซ้าย

(ค) ขั้นตอนหลังจากนำ RDF มาเปลี่ยนเป็นชุดตัวเลขไบนารี



(ง) ภาพรวมเมื่อทำงานร่วมกับ GPU

ภาพที่ 3.4 ภาพรวมการแปลงข้อมูลออนโทโลยีเป็นข้อมูลเข้าประมวลผลคิวรี่แบบขนาน

จากภาพสรุปได้ว่า ระบบทำการประมวลผลการทำงานบน 2 ส่วน ได้แก่ส่วนที่อยู่บนหน่วยประมวลผลกลาง และส่วนที่อยู่นบนหน่วยประมวลกราฟิกส์ โดยเริ่มจากผู้ใช้ป้อนคิวรี่เข้าไป ซึ่งจะถูกแปลงภายในฝั่งซีพียู และในขณะเดียวกันข้อมูลออนโทโลยีจะถูกแปลงเป็นลักษณะ RDF เช่นกัน จากนั้นคิวรี่ที่แปลงแล้วจะถูกส่งไปยังหน่วยประมวลกราฟิกส์เพื่อค้นหา และเมื่อพบคำตอบก็จะนำคำตอบออกจากหน่วยประมวลกราฟิกส์และมาแปลงกลับเพื่อแสดงผลต่อไป

3.2.1 **การแปลง RDF เป็นชุดตัวเลขไบนารี่**

การจัด RDF ด้านการท่องเที่ยวเชิงสุขภาพเป็นชุดข้อมูลที่แต่ละบรรทัดประกอบด้วย 3 ตัวหรือเรียกว่า Triple โดยเรียงจาก

Subject Predicate Object

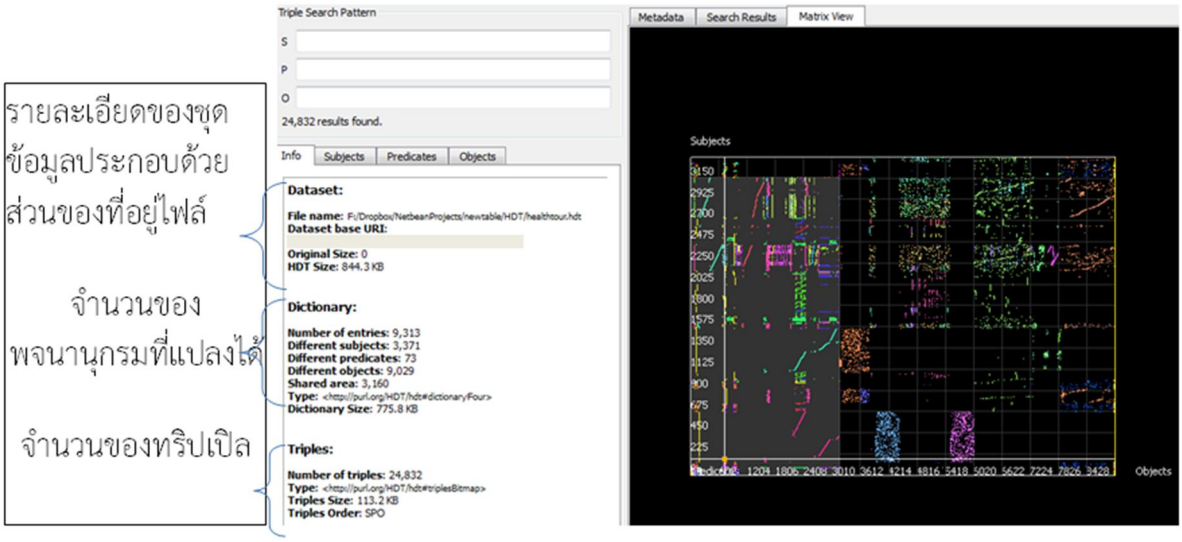
ซึ่งหมายถึงประธาน กริยา และกรรม ตามลำดับ ตัวอย่างเช่น โรงแรม ฮิลตัน มีสิ่งอำนวยความสะดวกห้องอบไอน้ำ โดยประธานคือ โรงแรม ฮิลตัน กริยาคือ มีสิ่งอำนวยความสะดวก และกรรมคือห้องอบไอน้ำ

ในโครงสร้าง HDT (Header-Dictionary-Triple) [33] มีการจัดเก็บข้อมูลบรรทัดดังกล่าวเป็นตัวเลข โดยมี unique ID สำหรับประธาน กริยา และกรรมตามลำดับ ถ้ามีหลาย Triple ก็จะมีหลายบรรทัดนั่นเอง ในกรณีทั่วไปประธานหนึ่งตัวอาจจะมีหลายๆ กริยา และกริยาหนึ่งตัวอาจจะมีหลายกรรม ดังนั้นจึงสามารถจัดกลุ่มโดยจัดความสัมพันธ์แบบต้นไม้ โดยทำการจัดกลุ่มของประธานเข้าด้วยกัน สำหรับแต่ละประธานก็จัดกลุ่มกริยาเข้าด้วยกัน และจัดเก็บเป็นลิสต์ ดังนี้

Subject (กลุ่ม Predicate ของ Subject นั้น (กลุ่ม Object ของ Predicate และ Subject นั้น))

จากนั้นในงานวิจัยนี้ ได้นำเสนอการแทนค่าชุดข้อมูลข้างต้นด้วย ตัวเลข ID ของประธาน กริยา กรรม ตามตำแหน่ง และใช้อะเรย์ของบิต 0-1 หรือไบนารี่อีกชุดเพื่อกำหนดตำแหน่งเริ่มต้นใหม่ของประธาน กริยา กรรมเป็นต้น

ดังนั้นทั้งประโยคคิวรี่ และชุดข้อมูลหลัก จะต้องถูกแปลงเป็นข้อมูลในรูปแบบเดียวกันนี้เพื่อค้นหา จากนั้นใช้อัลกอริธึมการค้นหาแบบใดก็ได้ที่รันบนหน่วยประมวลผลกราฟิกส์ช่วยค้นหาแบบขนาน ซึ่งในที่นี้ได้ทดสอบกับการค้นหาแบบบรูซฟอร์ซ (Brute force) นั่นคือการเปรียบเทียบเลข ID โดยดูตำแหน่งจากเลขไบนารีของชุดข้อมูลหลัก และเปรียบเทียบเลข ID ที่สัมพันธ์กันกับเลข ID ของประโยคที่ต้องการคิวรี่ ภาพที่ 3.5 แสดงการแปลงจากข้อมูล RDF เป็น HDT



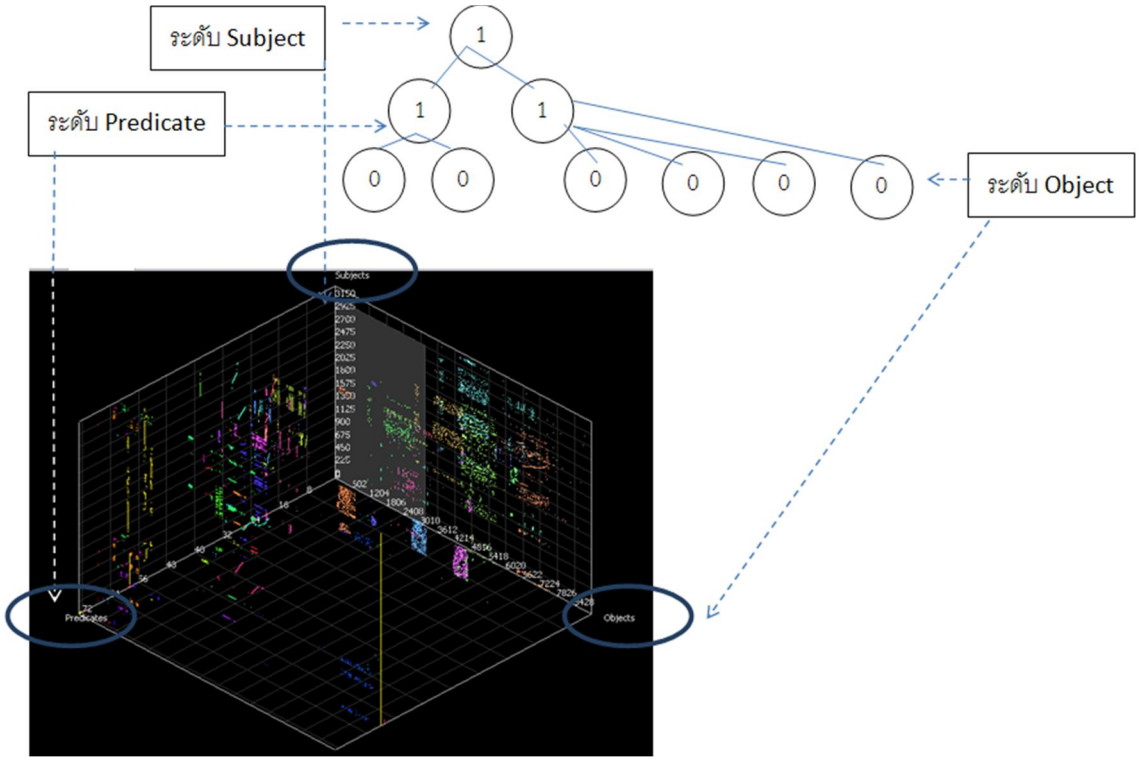
ภาพที่ 3.5 ข้อมูลที่ได้จากการแปลงแฟ้มออนโทโลยี RDF เป็น HDT

ขั้นตอนการแปลงชุดข้อมูลซึ่งเป็นการเตรียมชุดข้อมูลนั้น เป็นการแปลงชุดข้อมูลให้อยู่ในรูปต้นไม้ในการจัดเก็บแบบลิสต์ และอะเรย์ 1-0 (รวมสองอย่างนี้ เรียกว่า 1-0 Tree) เพื่อจะได้นำผลลัพธ์ที่ได้เข้าสู่กระบวนการค้นหาต่อประธาน 1 ตัวต่อไป การทำงานประกอบไปด้วยขั้นตอนต่อไปนี้

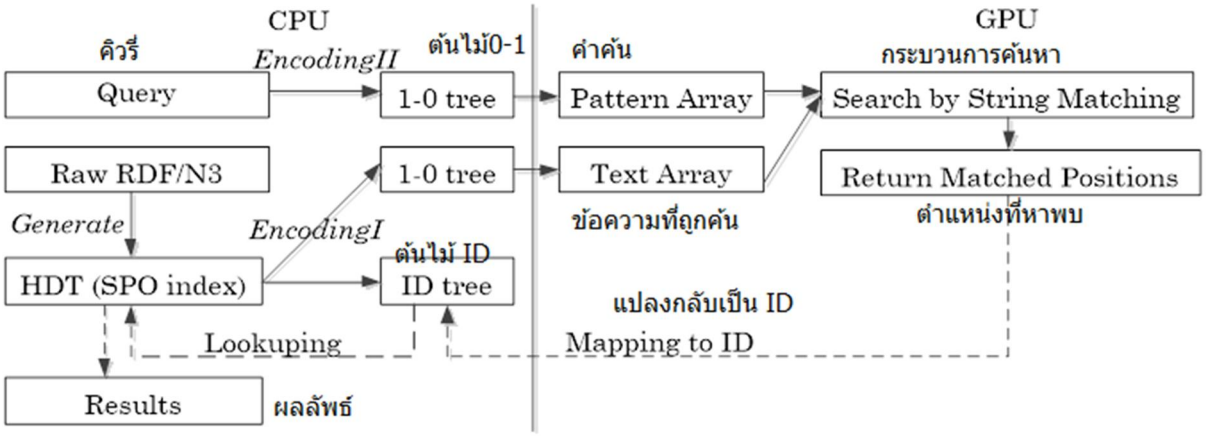
1) ตำแหน่งของชุดข้อมูลที่เป็นประธานให้แปลงเป็น 1
2) ตำแหน่งของชุดข้อมูลที่เป็นกริยาให้แปลงเป็น 1
3) ตำแหน่งของชุดข้อมูลที่เป็นกรรมให้แปลงเป็น 0

จากภาพ 3.6 สามารถเทียบขั้นตอนการแปลงเป็นต้นไม้ของโครงงาน (บน) ได้กับโครงสร้างเดิมของ HDT (ล่าง) ได้ คือ ระดับรากของต้นไม้คือ Subject ชุดเลข 1 สามารถเทียบได้กับแกนของ ของ HDT ในขณะที่ชุดเลข 1 ถัดมาเทียบได้กับแกนของ Predicate ของ HDT และชุดเลข 0 เทียบได้กับแกน Object ของ HDT

โครงสร้าง HDT จะมีโครงสร้าง 3 ส่วนได้ ได้แก่ Header, Dictionary, และ Triples ซึ่ง Triple นี้อยู่ในรูป S, P,O (Subject, Predicate, Object) ที่จะถูกนำมาใช้สร้างต้นไม้ 1 และ 0 ดังภาพ 3.6 โดยกำหนดให้ Subject S เป็นรากของต้นไม้ จากนั้นระดับถัดไปเป็น Predicate P และระดับต่อไปเป็น Object O ซึ่งจะมีต้นไม้หลายต้นก็ได้ ต้นไม้นี้เองจะถูกมองเป็นสายอักขระของ ID ที่สัมพันธ์กับสายอักขระของเลข 0-1 โดย ID จะหมายถึงรหัสตัวแทนของแต่ละ Subject Predicate และ Object

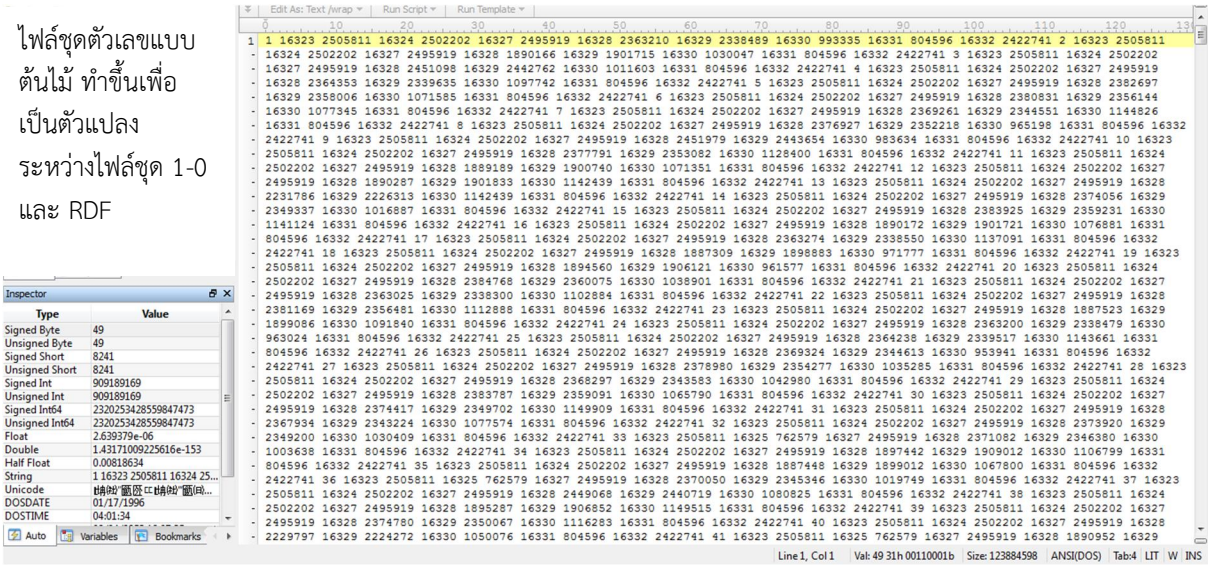


ภาพที่ 3.6 ชุดข้อมูลที่ได้หลังจากแปลงตัวเลข SPO ของ HDT

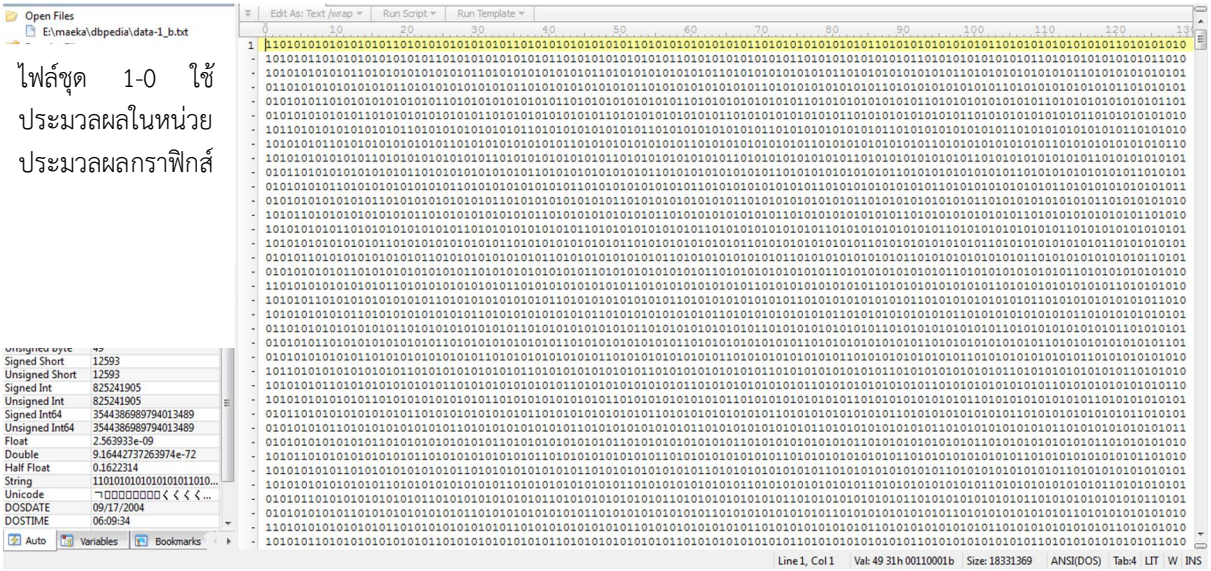


ภาพที่ 3.7 รายละเอียดขั้นตอนการแปลงและการค้นหา

ภาพที่ 3.7 แสดงขั้นตอนการแปลงและการค้นหา โดยเพิ่มรายละเอียดการแปลงข้อมูล ที่เป็นการเปลี่ยนรูปแบบการแทนค่าข้อมูลเพื่อให้สามารถนำเข้าสู่หน่วยประมวลผลกราฟิกส์เพื่อค้นหาได้เร็ว โดยคิวรี่จะถูกแปลงเป็น RDF ซึ่งจากนั้นจะถูกแปลงให้อยู่รูปของ HDT และแปลงเป็นสายอักขระของ ID และแปลงเป็นรูปแบบต้นไม้ 1 และ 0 (ดังภาพที่ 3.8 และภาพที่ 3.9) และนำเข้าไปในหน่วยความจำของหน่วยประมวลผลกราฟิกส์เพื่อการค้นหาต่อไป ดังนั้นการค้นหาจะเป็นการค้นหาแบบการค้นสายอักขระ 0-1 ซึ่งทำให้การค้นหาได้เร็วและจะใช้อัลกอริธึมค้นหาแบบใดก็ได้มาทำงาน



ภาพที่ 3.8 แฟ้มข้อมูลชุดตัวเลขของ Subject (Predicate(Object))



ภาพที่ 3.9 แฟ้มข้อมูล 1- 0 ที่ได้รับการแปลงจากแฟ้มชุดตัวเลข

3.2.2 **การค้นหาแบบขนานด้วยวิธีบรูซฟอร์ซของหน่วยประมวลผลกราฟิกส์**

การค้นหาบนหน่วยประมวลผลกราฟิกส์เกิดขึ้นเมื่อได้รับประโยคคิวรี่จากผู้ใช้งานผ่านหน้าจอและมีการแปลงประโยคคิวรี่นี้ ใช้เทียบกับชุดข้อมูลที่มีการแปลงไว้แล้ว จากนั้นทำการคัดลอกข้อมูลจากซีพียู ไปยังหน่วยประมวลผลกราฟิกส์เพื่อการค้นหาแบบขนาน

วิธีการค้นหาแบบขนานด้วยวิธีบรูซฟอร์ซของหน่วยประมวลผลกราฟิกส์นั้นแตกต่างจากการค้นหาผ่านหน่วยประมวลผลกลางที่เป็นการค้นหาแบบลำดับ โดยการค้นหาแบบลำดับเป็นการค้นโดยเลื่อนตำแหน่งระหว่างชุดข้อมูล กับประโยคคิวรี่ทีละลำดับ และสามารถค้นหาแบบต่อเนื่องจากหัวชุดข้อมูลไปยังท้ายชุดข้อมูลโดยใช้เทรดแบบเดี่ยว

การค้นหาแบบขนานด้วยวิธีบรูซฟอร์ซของหน่วยประมวลผลกราฟิกส์นั้น ให้มองว่าหน่วยประมวลผลกราฟิกส์สามารถทำแบ่งการประมวลผลเป็นแบบกริดได้ โดยแต่ละกริดสามารถแบ่งออกเป็นบล็อก โดยในแต่ละบล็อกจะมีเทรดหลายเทรด บล็อกละเท่า ๆ กัน ที่มีข้อจำกัดคือมีพื้นที่เก็บข้อมูลน้อยมากเมื่อเทียบกับเทรดของหน่วยประมวลผลกลางแต่สามารถทำงานพร้อมกันได้จำนวนหลายร้อยเทรด ดังนั้นการทำงานที่เรียบง่ายที่สุดคือ ส่งคำสั่งให้โปรแกรมแบ่งสายชุดข้อมูลตามจำนวนบล็อกและเทรดที่มากที่สุดเท่าที่หน่วยประมวลผลกราฟิกส์รุ่นนั้น ๆ จะสามารถบรรจุข้อมูลได้ แล้วจึงให้เทรดการค้นหาพร้อม ๆ กัน

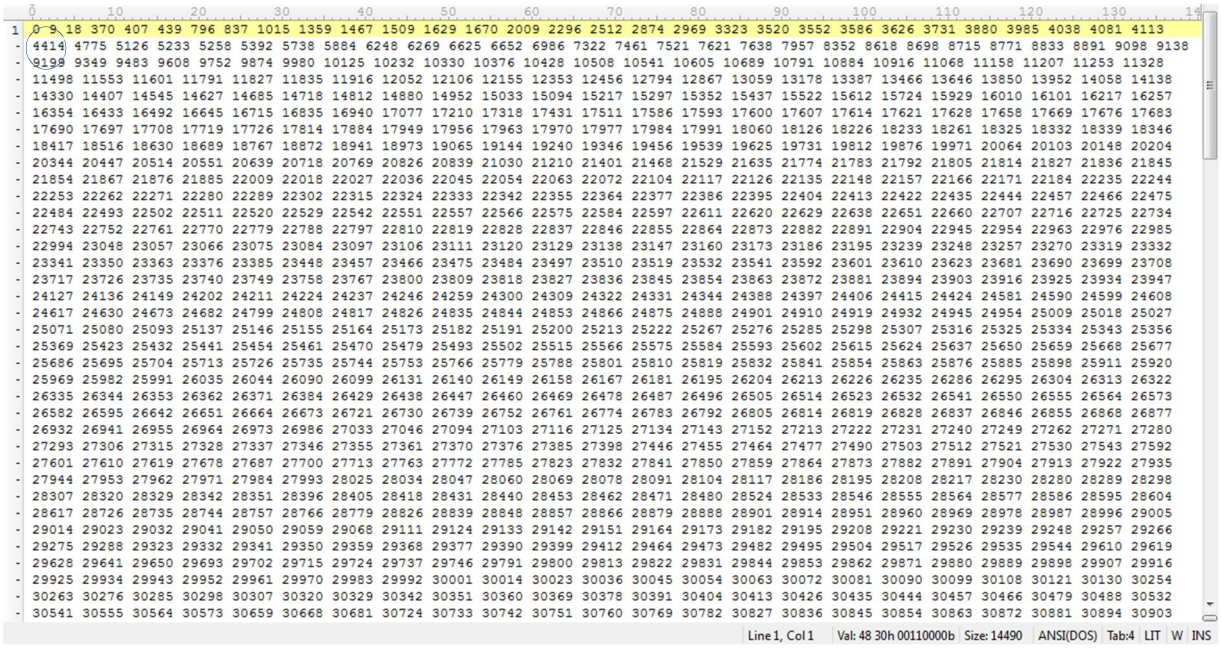
การค้นหาภายในเป็นแบบระบุตำแหน่งคงที่ สามารถแบ่งตามลำดับของชุดการแปลงแบบ Subject, Predicate, Object ได้โดยขั้นตอนต่อไปนี้

1) การค้นหา Subject มี 1 แบบ
   คือเลข 11 หมายถึงค้นหาตำแหน่งของเลข 1 ที่ตามด้วยเลข 1 ที่เป็นตำแหน่งของ Predicate
2) การค้นหา Predicate มี 1 แบบ
   คือเลข 10 หมายถึงค้นหาตำแหน่งของเลข 1 ที่ตามด้วยเลข 0 ที่เป็นตำแหน่งของ Object

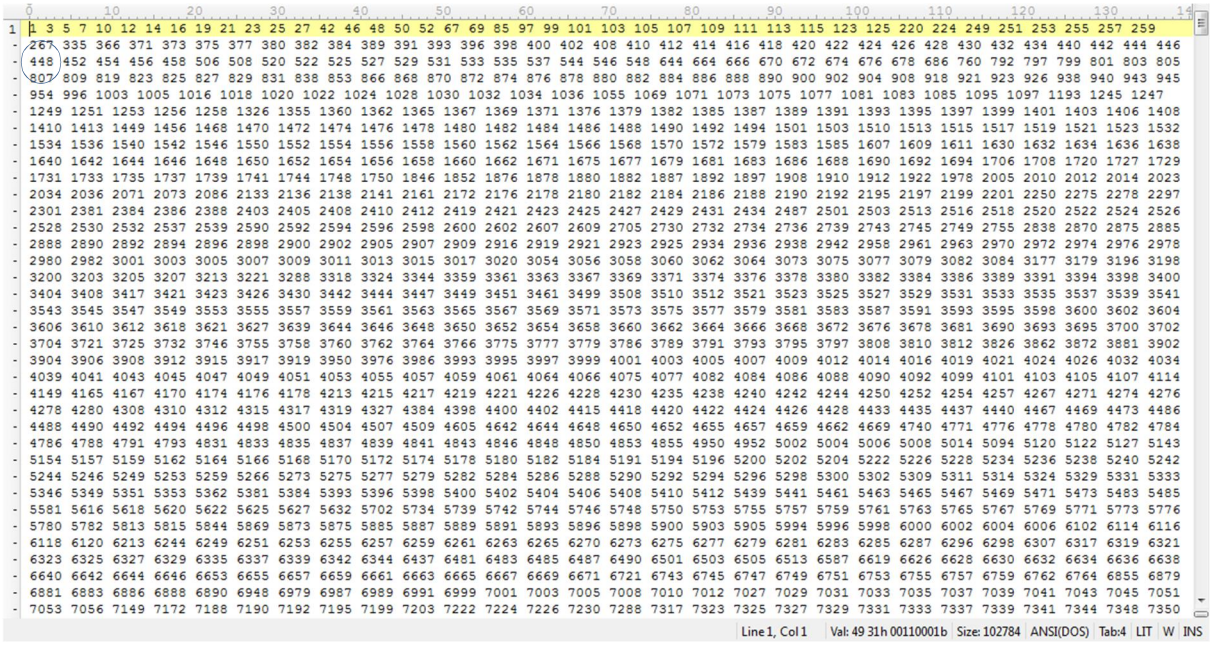
3) การค้นหา Object มี 3 แบบ
คือ การค้นหาเลข 00 คือลำดับของ Object ที่ตามด้วย Object ใน Predicate เดียวกันและการค้น แบบ 01 คือการหา Object ที่ตามด้วย Predicate ใน Predicate เดียวกัน หรือการค้นหา Object ที่ตามด้วย Subject ชุดใหม่ และสุดท้ายเป็นการค้นหาค่า Object ท้ายแฟ้มข้อมูลที่ตามด้วยเครื่องหมายจบแฟ้มข้อมูล

ในการคืนค่าคำตอบจากการคิวรี่ที่ได้จากหน่วยประมวลผลกราฟิกส์ จะได้ชุดของตำแหน่งตามลำดับของ Subject, Predicate และกลุ่มของ Object ในที่นี้ยกตัวอย่างการคืนค่าทั้งหมดได้ดังต่อไปนี้ Subject, Predicate และกลุ่มของ Object ในที่นี้ยกตัวอย่างการคืนค่าทั้งหมดได้ดังภาพที่ 3.10-ภาพที่ 3.13 โดยที่การค้นหาบนหน่วยประมวลผลกราฟิกส์เกิดขึ้น เมื่อได้รับประโยคคิวรี่จากผู้ใช้งานผ่านหน้าจอและมีการแปลงประโยคสั้นนี้แบบทันที เทียบกับชุดข้อมูลที่มีการแปลงไว้แล้ว จากนั้นทำการคัดลอกจากโฮสต์ไปยังดีไวส์ คือหน่วยประมวลผลกราฟิกส์ ที่เป็นการค้นหาแบบขนานบน

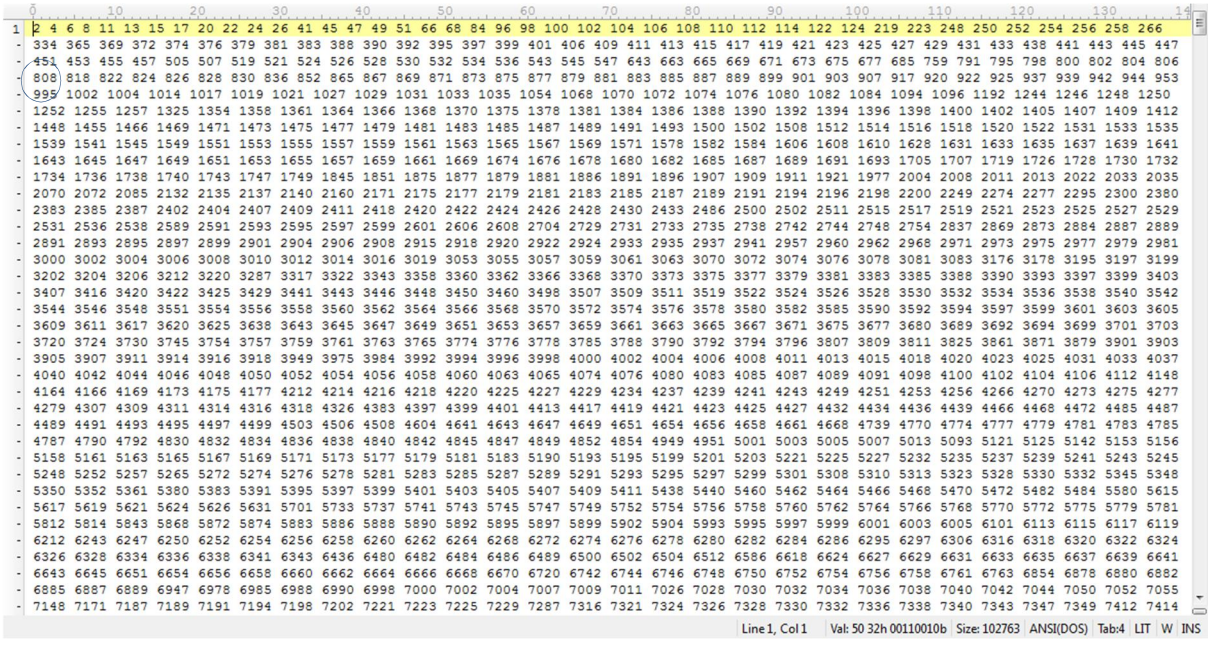
ข้อมูลที่ได้คืนกลับมาจะเป็นสายอักขระของ ID ตามตำแหน่งพอค่า ‘11’ (ภาพ 3.10) ซึ่งเป็นตำแหน่ง Suject จากนั้นจะนำเอา Subject นี้มาเทียบกับ Subject ที่ผู้ใช้ต้องการหา จากนั้นจะไปหา Predicate ที่สัมพันธ์กับ Subject ID นี้ จากนั้นจะไปหา Object ที่สัมพันธ์กับ Predicate ที่ผู้ใช้ต้องการส่งคืนลิสต์ของ Oject ID ที่ต้องการคืนไป เป็นต้นส่งคืนกลับผู้ใช้ และทำการแปลงข้อมูลกลับเป็นคำตอบเพื่อแสดงผล



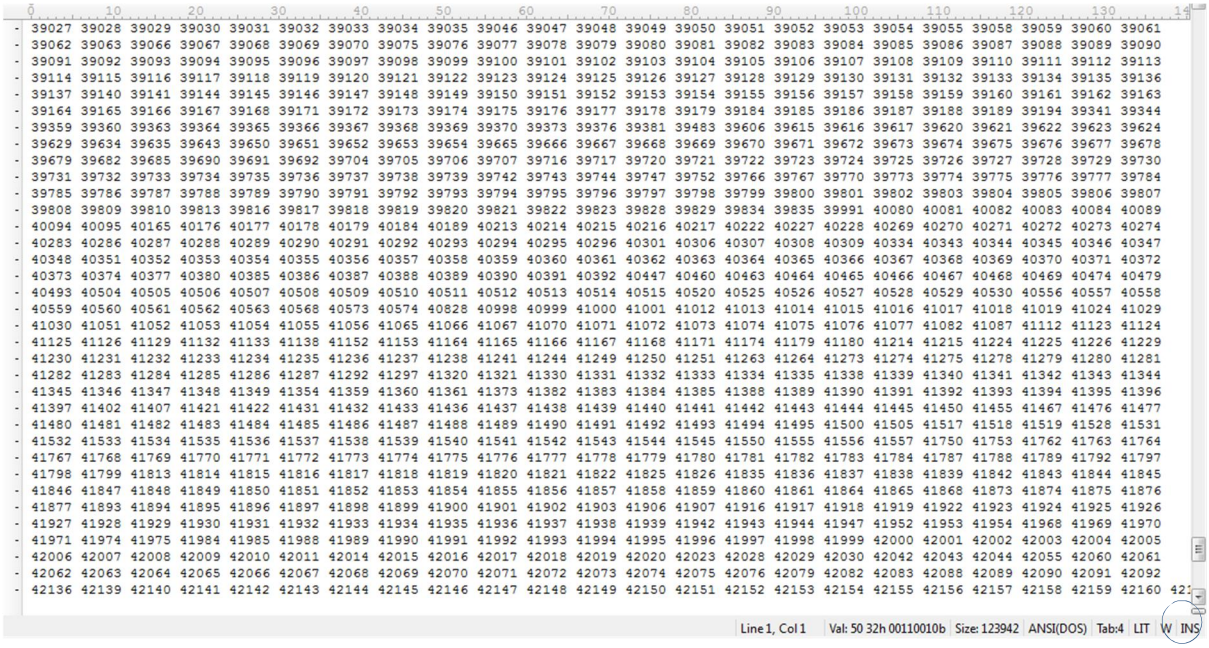
ภาพที่ 3.10 การคืนค่าของ Subject ID ตามจำแหน่งที่เจอ “11” ที่สามารถตรวจสอบได้จากตำแหน่งแรก



ภาพที่ 3.11 การคืนค่าของ Predicate ID “10”ที่สามารถตรวจสอบได้จากตำแหน่งแรกที่เป็นลำดับที่สอง



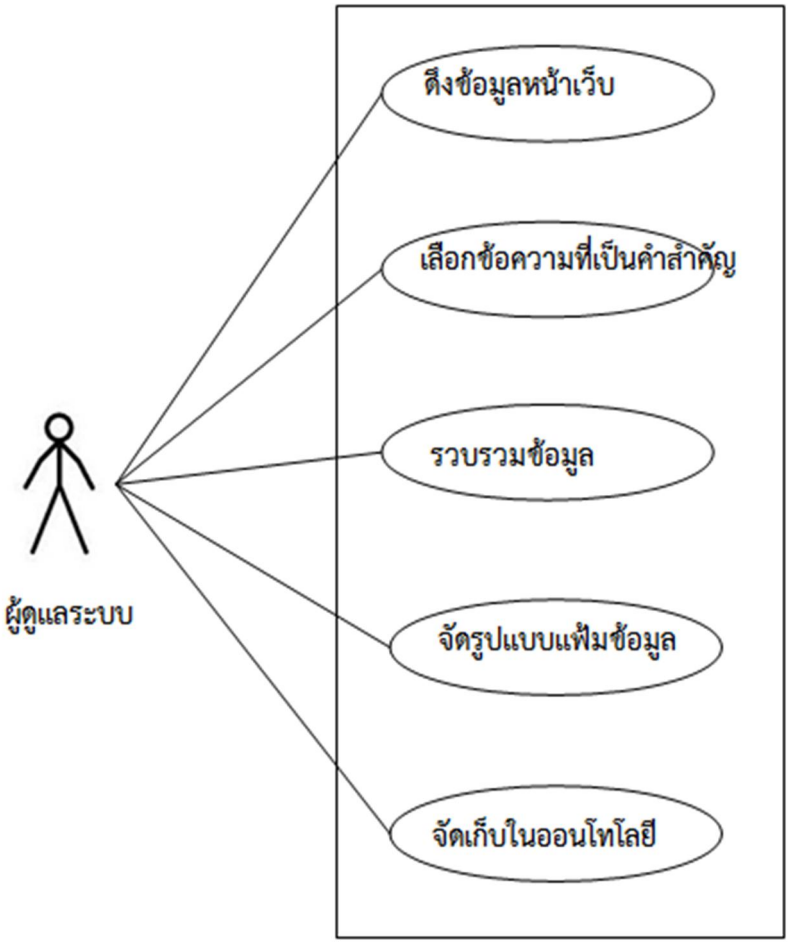
ภาพที่ 3.12 การคืนค่าของ Object ID ตาม “01” ที่สามารถตรวจสอบได้จากตำแหน่งแรกที่เป็นลำดับที่สาม



ภาพที่ 3.13 การคืนค่าของ Object ID ตาม “00” ที่สามารถตรวจสอบได้จากตำแหน่งสุดท้าย และสามารถคำนวณค่าสุดท้ายของ Object หลังสุดได้

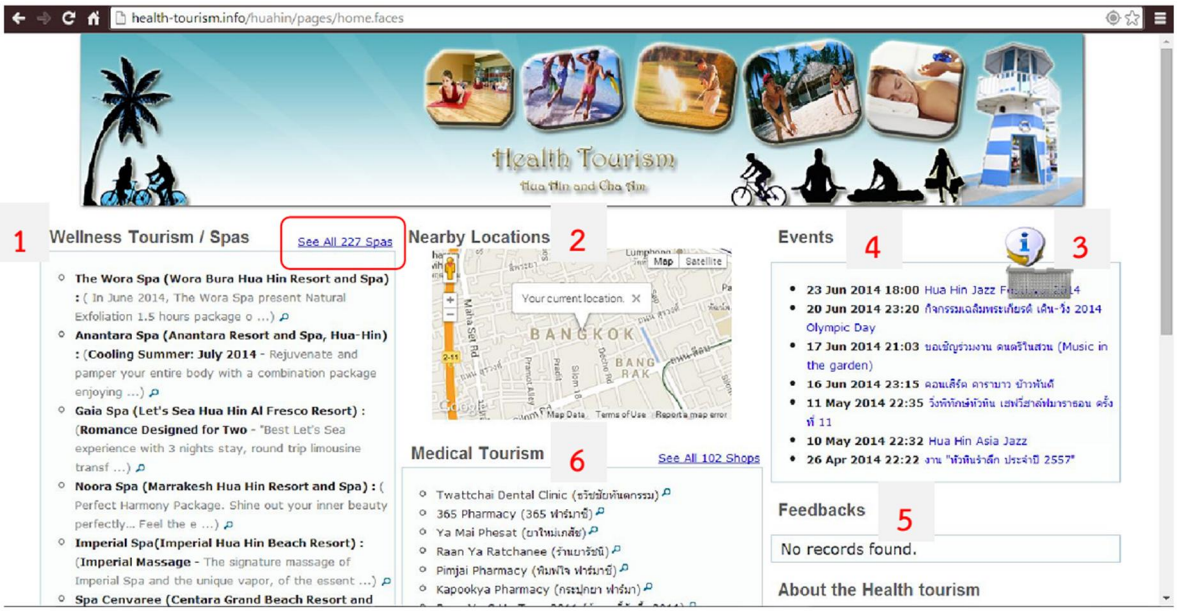
ในขั้นสุดท้ายการแปลงค่าคืนเพื่อส่งคำตอบให้ผู้ใช้สามารถทำได้โดยการเทียบตำแหน่งที่พบกับชุดตัวเลขในหน่วยประมวลผลกลาง แล้วจึงจับคู่กับลำดับตาม HDT เพื่อหาข้อความของ Triple ที่ต้องการออกมา แล้วจึงแปลงมาเป็นคำตอบบนเว็บแอพพลิเคชัน

3.3 Use Case Diagram และหน้าจอผู้ใช้งาน



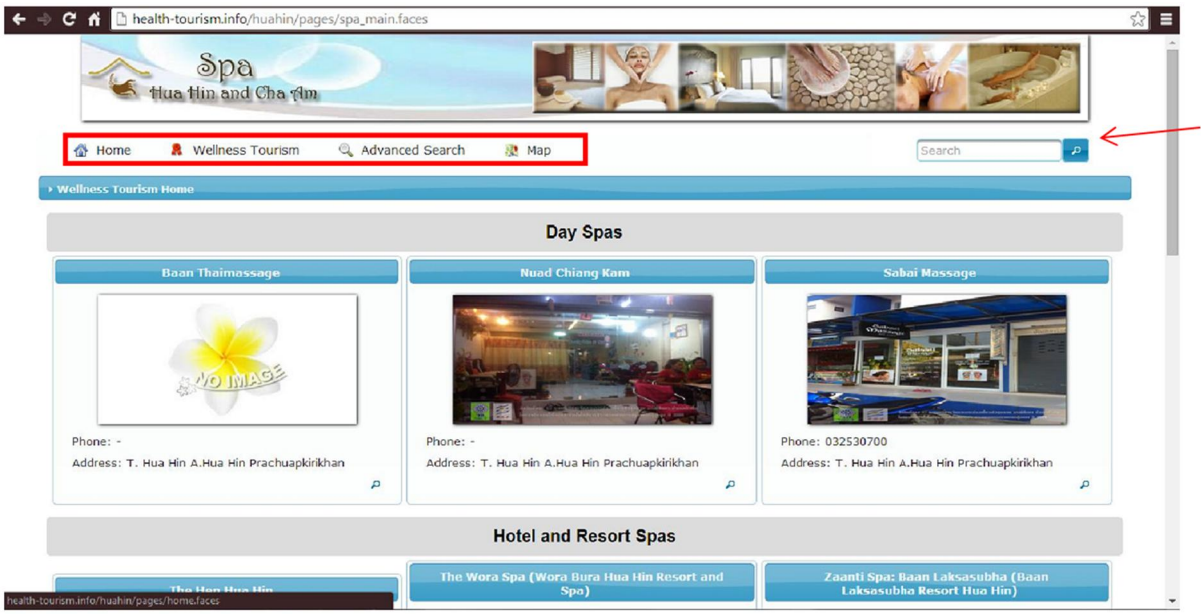
ภาพที่ 3.14 หน้าที่ของผู้ดูแลระบบ

ภาพที่ 3.14 แสดงหน้าที่ของผู้ดูแลระบบที่ทำหน้าที่นำข้อมูลจากเว็บไซต์มาเพิ่มในออนโทโลยี โดยที่ผู้ดูแลระบบมีการทำงานในแต่ละขั้นตอนเพื่อทำการเตรียมฐานความรู้เพื่อให้ผู้ใช้ระบบที่เป็นนักท่องเที่ยวทำการค้นหาข้อมูลเพื่อประโยชน์ในการประชาสัมพันธ์ด้านการท่องเที่ยวเชิงสุขภาพ



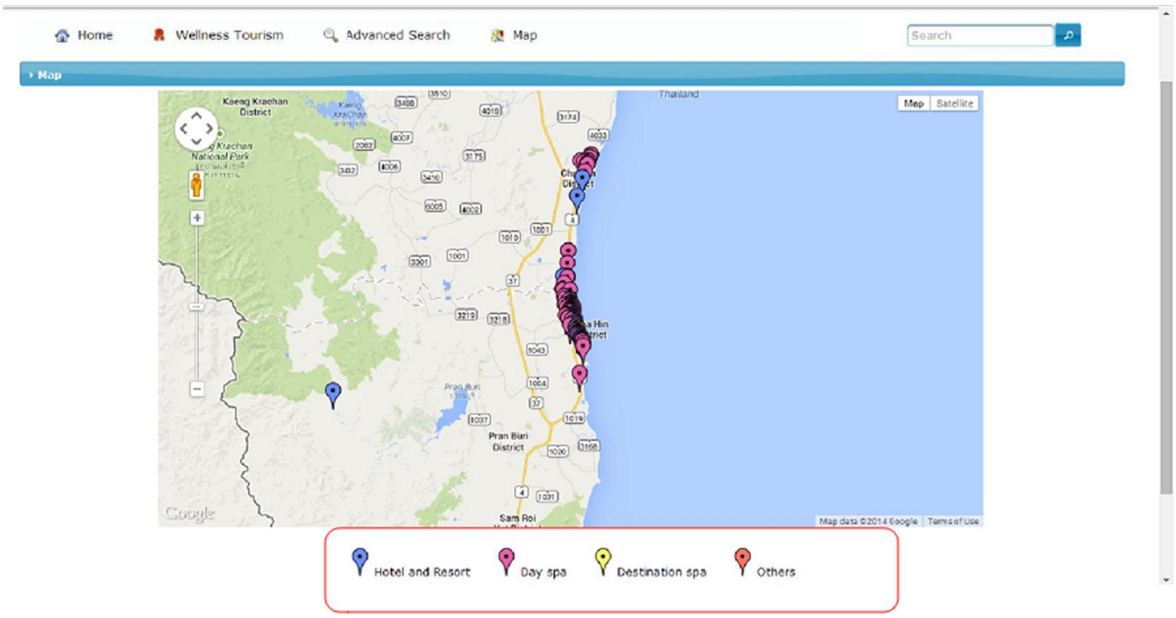
ภาพที่ 3.15 หน้าแรกของเว็บไซต์

ภาพที่ 3.15 แสดงหน้าจอหลักสำหรับการแสดงข้อมูล ผู้ใช้งานสามารถกดปุ่ม See All ที่อยู่ทางด้านขวาบนของจอภาพ ซึ่งจะทำให้ได้หน้าจอดังแสดงในภาพที่ 3.16



ภาพที่ 3.16 หน้า Wellness Tourism

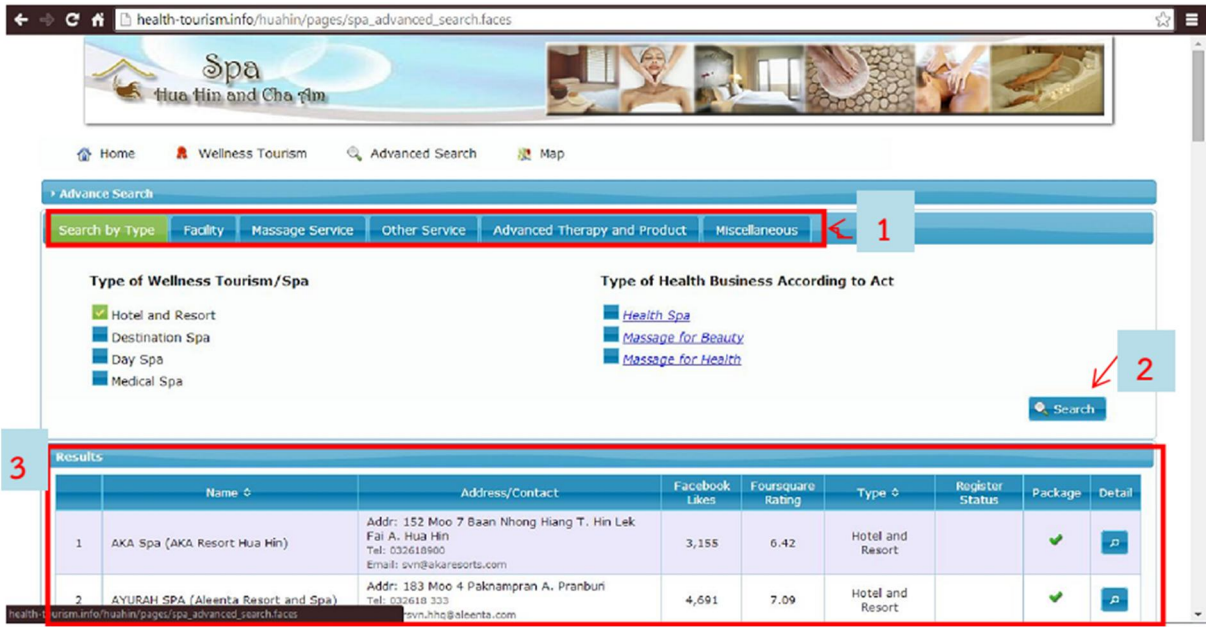
ภาพที่ 3.16 แสดงประเภทของสปา 3 ประเภท คือ Day Spa, Hotel and Resort Spa, Destination Spa, Medical Spa หมายเลข 1 ใช้ค้นหาร้านสปาโดยใช้คำสำคัญ (Keyword) และในกรอบสีแดงมีปุ่มเพื่อกดกลับไปหน้า HOME, Wellness Tourism, Advanced Search, Map การแสดงสถานบริการเหล่านี้จะทำการแสดงแบบสุ่มมา 3 สถานบริการในแต่ละหมวด



ภาพที่ 3.17 หน้าเว็บเมื่อกดเลือกที่ Map

จากเมนูหลักเมื่อเลือกแทป Map จะเป็นการเรียกหน้าเว็บที่แสดงแผนที่ ดังแสดงในภาพที่ 3.17 โดยที่ในการแสดงแผนที่นั้น สีของหมุดที่ปักหมายถึงประเภทของสถานบริการที่ต่างกัน (Hotel and Resort, Day Spa, Destination Spa, Others) นอกจากนี้ผู้ใช้งานสามารถค้นหารายละเอียดลักษณะของสปาในแบบที่ผู้ใช้ต้องการด้วยการเลือกแทป Advanced Search ทำให้ได้หน้าจอดังภาพที่ 3.18

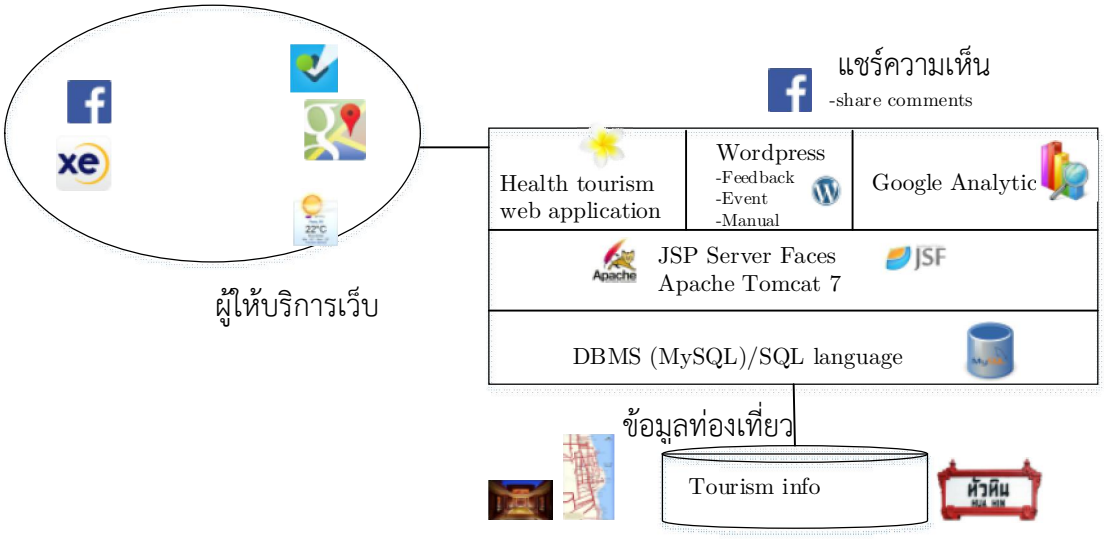
หากต้องการดูแผนที่ว่ามีร้านสปาอยู่ตำแหน่งใดบ้าง ให้เลือกแทป Map จะนำมายังหน้าถัดไปดังภาพที่ 3.17 ภาพที่ 3.17 ในการแสดงแผนที่นั้นสีของหมุดที่ปักนั้นจะหมายถึงประเภทสถานบริการทีต่างกัน ดังในกรอบด้านล่าง



ภาพที่ 3.18 หน้าเว็บเมื่อกดเลือก Advanced Search

ในภาพที่ 3.18 ประกอบด้วยการค้นหาตามรายละเอียดต่างๆ ตามแทป (1) ในที่นี้แทปต่างๆ ให้ใช้กำหนดคุณสมบัติในการค้นหา เมื่อคลิกจนขึ้นเครื่องหมายถูกแล้ว จะกดปุ่ม Search (หมายเลข 2) ผลของการค้นหาจะปรากฏในแทป Results (หมายเลข 3) โดยแสดงชื่อ (Name) ที่อยู่ (Address) จำนวน Like บนเฟสบุ๊คของสถานบริการ Rating ของ สถานบริการจาก Foursquare ประเภทสถานบริการ สถานบริการนั้นมี package หรือไม่ และปุ่มแว่นขยายเมื่อคลิกจะแสดง รายละเอียด (Detail) ของสถานบริการ รายละเอียดอื่นๆ สามารถอ่านเพิ่มเติมได้จากภาคผนวก ซ. แผนงานวิจัย

3.4 การเชื่อมต่อข้อมูลกับระบบอื่นๆ



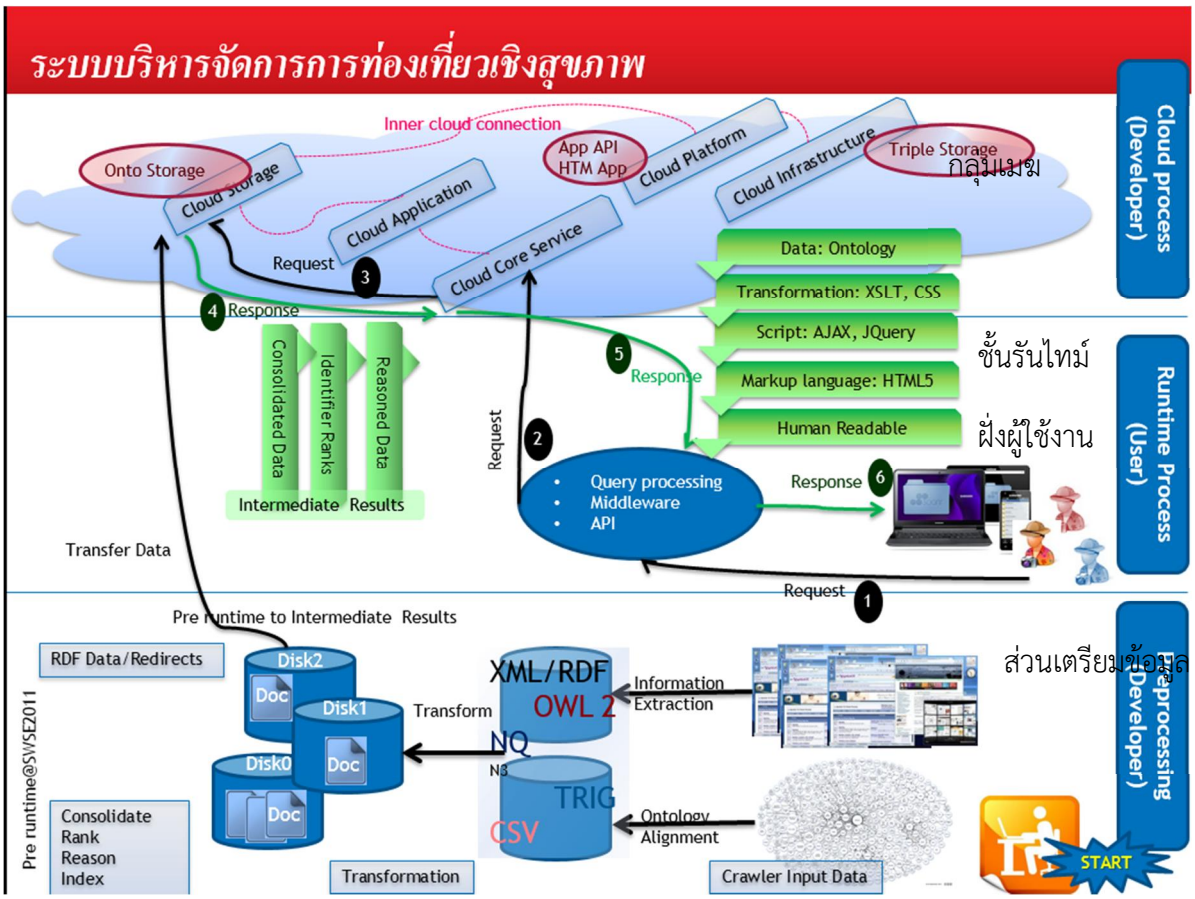
ภาพที่ 3.19 การเชื่อมต่อข้อมูลกับระบบข้อมูลอื่นๆ

สำหรับข้อมูลออนโทโลยีนั้น สามารถนำออกมาและนำไปใส่ในฐานข้อมูลแบบสัมพันธ์ เพื่อให้ง่ายต่อการดูแลรักษาต่อไป ในต้นแบบระบบนี้ ได้มีการนำฐานข้อมูลที่ได้ มาเชื่อมต่อกับส่วนอื่นๆ โดยมีโครงสร้างซอฟต์แวร์ที่เกี่ยวข้องในลำดับชั้น ได้แก่ ชั้นล่างสุดเป็นข้อมูลที่ได้เก็บมาจากเว็บไซต์และจากการสำรวจภาคสนาม โดยผ่านการคัดเลือกปรับแต่งรูปแบบให้เหมาะสม นำมาจัดเก็บในฐานข้อมูลที่ออกแบบไว้ ในตัวอย่างนี้ใช้ ฐานข้อมูล MySQL เชื่อมต่อด้วยซอฟต์แวร์ Apache Tomcat 7 และใช้ร่วมกับเฟรมเวิร์กในการพัฒนาเว็บได้แก่ JSP Server Faces ดังภาพที่ 3.19

ในส่วนของเว็บแอพพลิเคชันนั้น พัฒนาโดยมีหน้าจอหลักดังจะได้กล่าวต่อไป มีการเชื่อมต่อกับโมดูลต่างๆ เพื่อเก็บข้อมูลข่าวสาร (Event) ความเห็น (Feedback) ข้อเสนอแนะรวมทั้งการแชร์ข้อมูลผ่านเครือข่ายสังคม รวมทั้งคู่มือการใช้งานต่างๆ (Manual) ผ่าน Wordpress และ ใช้ Google analytics เพื่อการตรวจนับสถิติผู้เยี่ยมชม ในหมวดหมู่ต่างๆ

นอกจากนี้ ยังเชื่อมต่อกับระบบภายนอกผ่านเว็บเซอร์วิส เช่นจำนวน Like ของเฟสบุ๊ค Rating จาก Foursquare อัตราแลกเปลี่ยนจาก XE แผนที่จาก Google Map และพยากรณ์อากาศจาก Yahoo

3.5 โครงสร้างต้นแบบระบบบนกลุ่มเมฆ



ภาพที่ 3.20 ระบบบริหารจัดการการท่องเที่ยวเชิงสุขภาพต้นแบบ

จากภาพที่ 3.20 การทำงานของระบบบริหารจัดการการท่องเที่ยวเชิงสุขภาพประกอบด้วย 3 ระดับชั้น คือ ชั้นการจัดเตรียมข้อมูล (Preprocessing) ชั้นรันไทม์ (Runtime Process) และชั้นกลุ่มเมฆ (Cloud Process) โดยที่ชั้นจัดเตรียมข้อมูลนั้น ผู้ดูแลระบบเป็นผู้ดำเนินการเตรียมข้อมูลไว้ให้ผู้ใช้คิวรี่ ประกอบด้วยโปรแกรมแปลงข้อมูลมาใส่ออนโทโลยี จากนั้นนำไปเก็บไว้ที่เก็บข้อมูลบนกลุ่มเมฆ ร่วมกับแพลตฟอร์ม โครงสร้างและแอพพลิเคชัน เพื่อรอให้ผู้ใช้ร้องขอการคิวรี่จากชั้นรันไทม์มายังบริการในชั้นบนกลุ่มเมฆ การตอบกลับจากชั้นบนกลุ่มเมฆไปยังผู้ใช้แบ่งเป็นสองส่วนคือส่วนข้อมูลที่อยู่ในโครงสร้าง RDF และส่วนการแสดงผลทางหน้าเบราเซอร์

ภายในระดับชั้นของระบบการประมวลผลบนกลุ่มเมฆที่เกี่ยวข้องกับโครงงานย่อยที่ 1 แบ่งได้ 3 ระดับได้แก่ SaaS (Software as a Service) PaaS (Platform as a Service) และ IaaS (Infrastructure as a Service) มีรายละเอียดแต่ละชั้น ดังต่อไปนี้

1) **ชั้น IaaS** ประกอบด้วยเซิร์ฟเวอร์ ระบบเครือข่ายและส่วนเก็บออนโทโลยี
2) **ชั้น PaaS** ประกอบด้วยส่วนมิดเดิลแวร์ส่วนจัดการออนโทโลยี การใส่รายละเอียดหรือคำบรรยายหน้าเว็บและโปรแกรมส่วนรันไทม์

3) **ชั้น SaaS** ประกอบด้วย เว็บแอพพลิเคชัน HTML5 เว็บเซอร์วิสของผู้ให้บริการต่างๆเช่น สกุลเงิน สภาพอากาศ แผนที่ การท่องเที่ยวเชิงสุขภาพ และหน้าจอควบคุมแอพพลิเคชันเพื่อการส่งออกไปยังเครื่องมือต่างๆ

3.6 การประเมินความพึงพอใจ

คณะผู้วิจัยได้สร้างแบบสอบถามเพื่อสำรวจความพึงพอใจในด้านการให้บริการข้อมูล โดยทำการเก็บรวบรวมข้อมูลจากแบบสอบถามผู้ใช้งานระบบ และใช้ค่าทางสถิติที่ทำการวัดแนวโน้มเข้าสู่ส่วนกลางด้วยการวัดค่ากลางของข้อมูลโดยใช้ค่าเฉลี่ย (Mean) และวัดการกระจายของข้อมูลโดยใช้ส่วนเบี่ยงเบนมาตรฐาน (Standard Deviation)

ค่าตัวกลางเลขคณิต (Arithmetic Mean) หรือค่าเฉลี่ย

$$\boldsymbol{X = \frac{\sum_{i=1}^{n} x_i}{n}}$$

โดยที่ X แทนตัวกลางเลขคณิตหรือค่าเฉลี่ย

n แทนจานวนข้อมูลทั้งหมด

xi แทนข้อมูลดิบตัวที่ *i*

$\sum_{i=1}^{n} x_i$ แทนผลรวมทั้งหมดของข้อมูล

ค่าส่วนเบี่ยงเบนมาตรฐาน (Standard deviation)

$$\boldsymbol{SD = \sqrt{\frac{\sum_{i=1}^{N}(x_i - X)^2}{N}}}$$

โดยที่ SD แทนค่าเบี่ยงเบนมาตรฐานของกลุ่มตัวอย่าง

X แทนค่าเฉลี่ยของกลุ่มตัวอย่าง

x_i แทนค่าของข้อมูล

i แทน 1,2,3,...., N

N แทนค่าของจำนวนข้อมูล (กลุ่มตัวอย่าง)

คณะผู้วิจัยได้กำหนดเกณฑ์การให้คะแนนในแบบประเมินตามตารางที่ 3.1 เกณฑ์การให้คะแนนของแบบประเมินและกำหนดเกณฑ์การแปลความหมายข้อมูลและพิจารณาจากค่าตัวกลางเลขคณิต(ค่าเฉลี่ย) ดังแสดงในตาราง 3.2

ตารางที่ 3.1 เกณฑ์การให้คะแนนของแบบประเมิน

ระดับเกณฑ์	**ความหมาย**
5	ประสิทธิภาพของระบบอยู่ในระดับดีมาก
4	ประสิทธิภาพของระบบอยู่ในระดับดี
3	ประสิทธิภาพของระบบอยู่ในระดับปานกลาง
2	ประสิทธิภาพของระบบอยู่ในระดับน้อย
1	ประสิทธิภาพของระบบอยู่ในระดับควรปรับปรุง

ตารางที่ 3.2 เกณฑ์การแปลความหมายข้อมูลและพิจารณาจากค่าตัวกลางเลขคณิต (ค่าเฉลี่ย)

ระดับเกณฑ์	**ความหมาย**
4.50 – 5.00	ประสิทธิภาพของระบบอยู่ในระดับดีมาก
3.50 – 4.49	ประสิทธิภาพของระบบอยู่ในระดับดี
2.50 – 3.49	ประสิทธิภาพของระบบอยู่ในระดับปานกลาง
1.50 – 2.49	ประสิทธิภาพของระบบอยู่ในระดับน้อย
1.00 – 1.49	ประสิทธิภาพของระบบอยู่ในระดับควรปรับปรุง

บทที่ 4 ผลการวิจัย

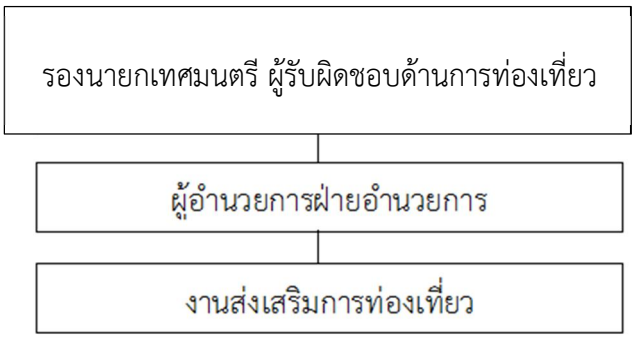
ผลการวิจัยดำเนินงานแบ่งเป็น 3 ส่วน ดังนี้ ส่วนแรกเกี่ยวกับการสำรวจลักษณะของซอฟต์แวร์ที่ใช้ในปัจจุบันของหน่วยงาน และข้อมูลที่หน่วยงานใช้อยู่ ส่วนที่ 2 เป็นผลการประเมินประสิทธิภาพด้านเวลาของระบบเว็บเชิงความหมาย การประมวลผลแบบขนาน ส่วนที่ 3 เป็นผลการประเมินประสิทธิภาพด้วยเวลาสำหรับการประมวลผลการดึงข้อมูล และค้นหาข้อมูลแบบขนาน ส่วนการประเมินผู้ใช้งานนั้นอยู่ในภาคผนวกของรายงานแผนงานวิจัย จึงไม่ขอกล่าวถึงในที่นี้

4.1 การสำรวจข้อมูลของเทศบาลด้านลักษณะหน่วยงาน ข้อมูลที่มีและซอฟต์แวร์ต่างๆ

คณะผู้วิจัยได้ สอบถามข้อมูลซอฟต์แวร์ที่ใช้ภายในหน่วยงานเทศบาล และการทำงานของหน่วยงาน แหล่งข้อมูลต่างๆ ของการท่องเที่ยวที่ทางเทศบาลใช้อยู่ ด้วยการสัมภาษณ์และแบบสอบถาม โดยรายละเอียดอยู่ในรายงานแผนการวิจัย หัวข้อ 4.1 และ 4.2 และในภาคผนวก ก. ของรายงานแผนการวิจัย

4.1.1 ข้อมูลเกี่ยวข้องกับหน่วยงานเทศบาลเมืองหัวหิน

ผู้ที่เกี่ยวข้องกับโครงงานนี้ในหน่วยงานของเทศบาลเมือง คือ นายกเทศมนตรีมอบหมายงานให้รองนายกเทศมนตรีด้านการท่องเที่ยว คือคุณมนตรี ชูภู่ เป็นผู้อนุมัติการประเมินระบบนี้



ภาพที่ 4.1 แผนผังหน่วยงานที่เกี่ยวข้องกับงานวิจัย

รองนายกเทศมนตรี ผู้รับผิดชอบด้านการท่องเที่ยว ได้มอบหมายฝ่ายอำนวยการ คือคุณเบญจวรรณ เมธาวิกุล มีหน้าที่ควบคุมดูแลและรับผิดชอบการปฏิบัติงานในหน้าที่ที่เกี่ยวข้องกับงานส่งเสริมการท่องเที่ยว มีหัวหน้างานคือ คุณอุกฤษฎ์ ป้อทองคำ มีพนักงาน 7 คน สำหรับงานส่งเสริมการท่องเที่ยว มีหน้าที่เกี่ยวกับ

- งานส่งเสริมและเผยแพร่การท่องเที่ยวของเทศบาลภายในประเทศ
- งานควบคุมกิจการบริการท่องเที่ยวภายในประเทศของเทศบาลและหน่วยงานอื่น
- งานต้อนรับ แนะนำ อำนวยความสะดวกและบริการนำเที่ยว รวมทั้งดำเนินธุรกิจการท่องเที่ยว

- งานประชาสัมพันธ์ ให้คำแนะนำเผยแพร่ความรู้เกี่ยวกับแหล่งท่องเที่ยวหรือกิจการ ที่เกี่ยวกับการท่องเที่ยวภายในจังหวัดและเทศบาล
- งานจัดทำรายงานและสถิติการเคลื่อนไหวเกี่ยวกับธุรกิจการท่องเที่ยว
- งานอื่นที่เกี่ยวข้องหรือตามที่ได้รับมอบหมาย

4.1.2 **สรุปข้อมูลที่ได้จากการสำรวจความต้องการและข้อมูลที่มีอยู่**

รายละเอียดของการสำรวจความต้องการซอฟต์แวร์จากหน่วยงานเทศบาลเมืองหัวหินสำหรับการสำรวจข้อมูลในพื้นที่นั้นได้ทำ 2 ครั้ง นั่นคือ ครั้งที่ 1 เมื่อวันที่ 23 สิงหาคม พ.ศ. 2556และได้กำหนดวันนัดหมายเพื่อพูดคุยรายละเอียดกันในเดือนกันยายน แต่ได้มีการเลื่อนเป็นครั้งที่ 2 วันที่ 4 ตุลาคม พ.ศ. 2556 ดังรายละเอียดดังต่อไปนี้

ระดับผู้บริหารนั้นมีความต้องการใช้ข้อมูลที่ผ่านการรวบรวมอย่างเป็นระบบ เนื่องจากการนำเสนอข้อมูลจากหน่วยงานต่อสาธารณชนนั้นควรมีรายละเอียดรอบด้านมากขึ้น

การสัมภาษณ์ผู้อำนวยการฝ่ายอำนวยการ และสำรวจจากเทศบาลเมืองหัวหิน หัวหน้างานส่งเสริมการท่องเที่ยว และเจ้าหน้าที่การท่องเที่ยวแห่งประเทศไทยในเขตเมืองหัวหินและพนักงานปฏิบัติการ คุณเกษมศักดิ์ ปัดออด มีดังแบบสำรวจความต้องการการใช้งานซอฟต์แวร์ของระบบการจัดการท่องเที่ยวเชิงสุขภาพ สรุปผลการสำรวจได้ดังนี้

- ระดับหน่วยงานส่งเสริมการท่องเที่ยวมีศูนย์บริการนักท่องเที่ยว 2 แห่งคือเทศบาลเมืองหัวหินและหอนาฬิกา
- การรวบรวมข้อมูลเป็นการทำงานแบบระบบมือ
- การค้นหาข้อมูลเกี่ยวกับที่พัก ร้านอาหาร สถานที่แจ้งภัย ส่วนมากใช้ระบบเดินสำรวจหรือค้นหาทางเว็บไซต์เฉพาะที่ ในขณะที่สถานพยาบาล สปา นวดแผนไทย อายุรเวช นั้นไม่เคยสำรวจ
- การติดต่อสื่อสารระหว่างหน่วยงานเทศบาลและที่พักชั่วคราวอาทิโรงแรมในพื้นที่มักใช้โทรศัพท์และการเดินเอกสาร สำหรับการใช้งานโปรแกรมจองโรงแรมหรือเว็บไซต์เกี่ยวกับการท่องเที่ยว เช่น Agoda, TripAdvisor, AtSiam นั้น ไม่เคยใช้ สำหรับโปรแกรมการค้นหาเว็บไซต์นั้นใช้กูเกิล ส่วนเครือข่ายสังคมออนไลน์นั้นใช้เฟสบุ๊คเพื่อกระจายข่าวสารเทศบาลและการท่องเที่ยว
- สำหรับเครื่องมือที่ใช้ค้นหาผ่านอินเทอร์เน็ตคือคอมพิวเตอร์ส่วนบุคคลที่เชื่อมต่ออินเตอร์เน็ตทั้งแลนและระบบไร้สาย
- การทำรายงานนำเสนอผู้บริหารประกอบด้วยเชื้อชาติของนักท่องเที่ยวและจำนวนผู้สอบถามเป็นรายวันรายเดือนและรายปีผ่านโปรแกรมไมโครซอฟต์เอ็กเซล

- สำหรับเอกสารเผยแพร่ข้อมูลจัดทำเป็นแผ่นพับสองภาษาคือภาษาไทยและภาษาอังกฤษ และมีการปรับปรุงประมาณปีละครั้ง
- สิ่งที่ต้องการจากระบบบริหารจัดการการท่องเที่ยวเชิงสุขภาพคือข้อมูลโรงแรมเบื้องต้น เพิ่มเติมจากการการลงพื้นที่เองและจากกูเกิล ประโยชน์ที่จะนำไปใช้นอกจากรายงาน สำหรับผู้บริหารคือ เป็นแหล่งข้อมูลสำหรับนักวิจัย นักเรียน นักศึกษา ในกรณีที่มีการร้องขอข้อมูลอย่างละเอียด
- ข้อมูลที่ร้องขอเพิ่มเติมคือหมายเลขโทรศัพท์และอีเมล์ของสถานทูต ข้อมูลสำนักงานตรวจคนเข้าเมืองเพื่อต่อวีซ่า
- ต้องการใช้ระบบบริหารจัดการการท่องเที่ยวเชิงสุขภาพ เพื่อรวบรวมข้อมูลร้อยละ 45 เพื่อตรวจสอบข้อมูลร้อยละ 45 ใช้ประกอบรายงานสำหรับผู้บริหารร้อยละ 10
- หากโปรแกรมเสร็จแล้วมีความต้องการอบรมเพื่อใช้งานโปรแกรม

นอกจากนี้ได้มีการสอบถามข้อมูลการวางแผนท่องเที่ยวจากสำนักงานการท่องเที่ยวแห่งประเทศไทย สำนักงานประจวบคีรีขันธ์วันที่ 23 สิงหาคม พ.ศ. 2556 ในเรื่องการวางแผนการท่องเที่ยวสำหรับนักท่องเที่ยวทั้งชาวไทยและชาวต่างประเทศซึ่งมีการเก็บข้อมูลนักท่องเที่ยวที่คล้ายคลึงกับเทศบาลเมืองคือเชื้อชาติและจำนวนนักท่องเที่ยวที่มาสอบถามรายวัน สำหรับข้อมูลรายได้และนักท่องเที่ยวโดยรวมของภาคกลางจะได้รับมาจากการท่องเที่ยวแห่งประเทศไทยสำนักงานใหญ่การวางแผนท่องเที่ยวในส่วนข้อมูลทั่วไปและกิจกรรมการท่องเที่ยวที่เผยแพร่นั้นเจ้าหน้าที่เป็นผู้รวบรวมขึ้นเองส่วนเครื่องมือค้นหาเว็บไซต์นั้นใช้งานผ่านกูเกิลสำหรับรายละเอียดของหน้าที่สำนักงาน กิจกรรมการท่องเที่ยว รายละเอียดนักท่องเที่ยวและแผนการท่องเที่ยวมีให้ตามแผ่นพับของ ททท.

4.1.3 **ข้อมูลจากหน่วยงานเพื่อตรวจสอบความถูกต้อง**

สำหรับสถานบริการสปาหรือนวดแผนโบราณที่เป็นเป้าหมายของงานวิจัย ในกรณีที่เป็นโรงแรมหรือที่พักขนาดใหญ่ที่มีอยู่ใน Agoda หรือ TripAdvisor นั้น จะถูกดึงเข้าสู่ระบบโดยโปรแกรมคอมพิวเตอร์ อย่างไรก็ตามยังมีสถานบริการริมชายหาดและสถานบริการเชิงสุขภาพขนาดเล็ก ซึ่งผู้อำนวยการฝ่ายอำนวยการได้กรุณาให้ความรู้ในไว้ดังนี้

- กรณีที่เป็นการนวดริมชายหาดของเอกชนนั้นสามารถตรวจสอบได้กับเทศกิจ
- กรณีที่เป็นร้านขนาดเล็ก จะมีการขึ้นทะเบียนกับกองสาธารณสุข สิ่งที่ได้คือชื่อร้านและที่อยู่
- สำหรับพิกัดที่ตั้งนั้น งานแผนที่ภาษีมีบันทึกไว้ในระบบพิกัดกริดแบบ UTM รายชื่อผู้เสียภาษีและที่อยู่

- เมื่อต้องการนำมาตรวจสอบกับระบบที่ผู้วิจัยเก็บไว้เพื่อให้ได้ชื่อร้านและพิกัดละติจูด/ลองจิจูด ต้องนำข้อมูลมาจากทั้ง 3 หน่วยงานนี้มาตรวจสอบร่วมกัน

4.2 ผลการประเมินประสิทธิภาพ

การวัดประสิทธิภาพด้านเวลามี 2 แบบ แบบแรกเป็นการวัดประสิทธิภาพด้านเวลาการค้นหาด้วยเว็บเชิงความหมายโดยค้นหาในออนโทโลยี และแบบที่สองเป็นการค้นหาสายอักขระด้วยการประมวลผลแบบขนานด้วยหน่วยประมวลผลกราฟิก

4.2.1 **เวลาการค้นหาด้วยเว็บเชิงความหมาย**

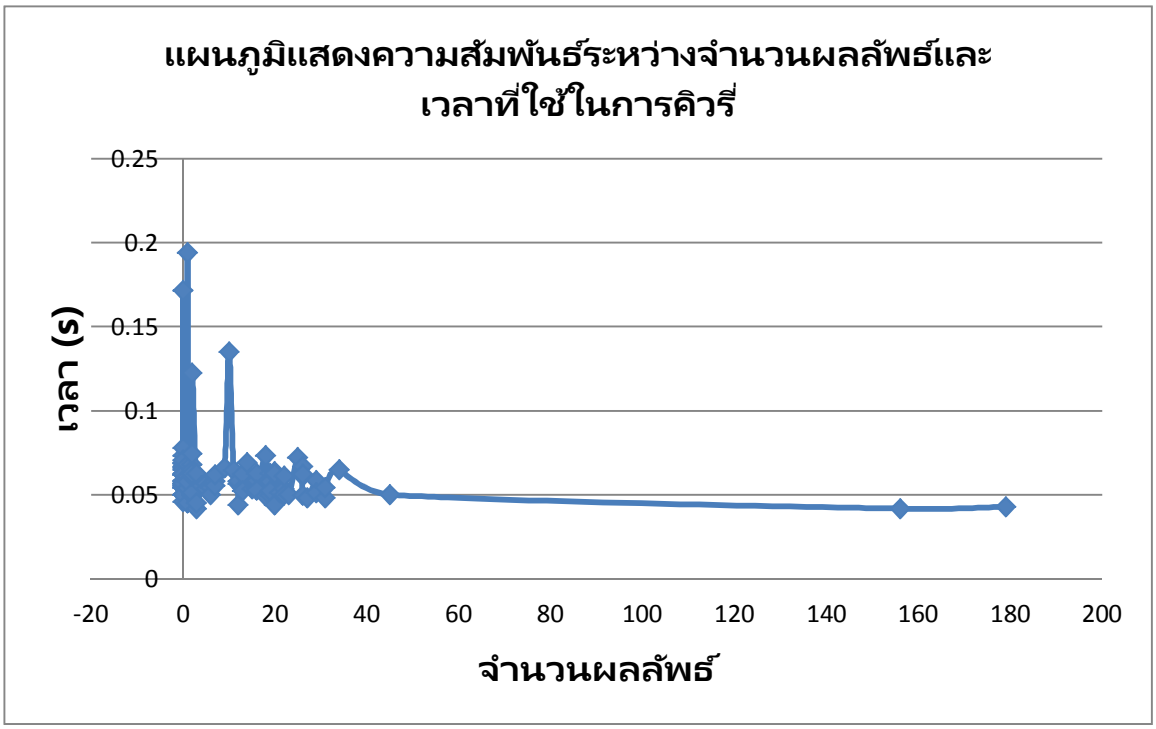
ในการทดสอบประสิทธิภาพ คณะผู้วิจัยได้ทดสอบวัดความเร็วคิวรี่จากจำนวนคิวรี่100 ตัวอย่าง ทั้งที่ไม่มีความซับซ้อน มีความซับซ้อนน้อยไปจนถึงตัวอย่างที่มีความซับซ้อนมากเพื่อวัดเวลาที่ใช้ในการคิวรี่ข้อมูล ตารางที่ 5.1 แสดงรายละเอียดข้อมูลออนโทโลยี

ตารางที่ 4.1 ข้อมูลออนโทโลยี

ข้อมูลทั้งหมดในออนโทโลยี	**จำนวน**
Axiom (ข้อกำหนดความสัมพันธ์)	24663
Logical axiom (ความสัมพันธ์เชิงตรรกะ)	21094
Class (คลาส)	229
Object Property (ความสัมพันธ์)	33
Data Property (คุณสมบัติ)	29
Individual (สมาชิกของคลาส)	3008
ข้อจำกัดความสัมพันธ์ในคลาส	**จำนวน**
SubClassOf (คลาสย่อย)	227
EquivalentClasses (คลาสเทียบเท่า)	19
DisjointClasses (คลาสที่ไม่เกี่ยวข้องกัน)	236
Hidden GCI (General Concept Inclusions)	13
ข้อจำกัดความสัมพันธ์ใน Object Property	**จำนวน**
SubObjectPropertyOf (ความสัมพันธ์ย่อย)	2
EquivalentObjectProperties (ความสัมพันธ์เทียบเท่า)	2
InverseObjectProperties (ความสัมพันธ์ที่แบบผันกลับ)	16
FunctionalObjectProperty (ความสัมพันธ์ระหว่างกันแบบหนึ่งต่อหนึ่ง)	2
TransitiveObjectProperty (ความสัมพันธ์ที่มีการส่งต่อไปเรื่อยๆ)	4
ObjectPropertyDomain (สมาชิกภายในเซตข้อมูลที่ใช้จับคู่กับอีกเซตหนึ่ง	33
ObjectPropertyRange (เซตข้อมูลที่ถูกจับคู่โดยความสัมพันธ์)	26
ข้อจำกัดความสัมพันธ์ใน Annotation	**จำนวน**
AnnotationAssertion (คำอธิบายประกอบคุณสมบัติ ความสัมพันธ์ และข้อมูล	271
ข้อจำกัดความสัมพันธ์ใน Individual	
ClassAssertion (การกำหนดสมาชิกในคลาส	3076
ObjectPropertyAssertion (การกำหนดความสัมพันธ์ของข้อมูล)	9943
DataPropertyAssertion (การระบุคุณสมบัติของข้อมูล)	7454

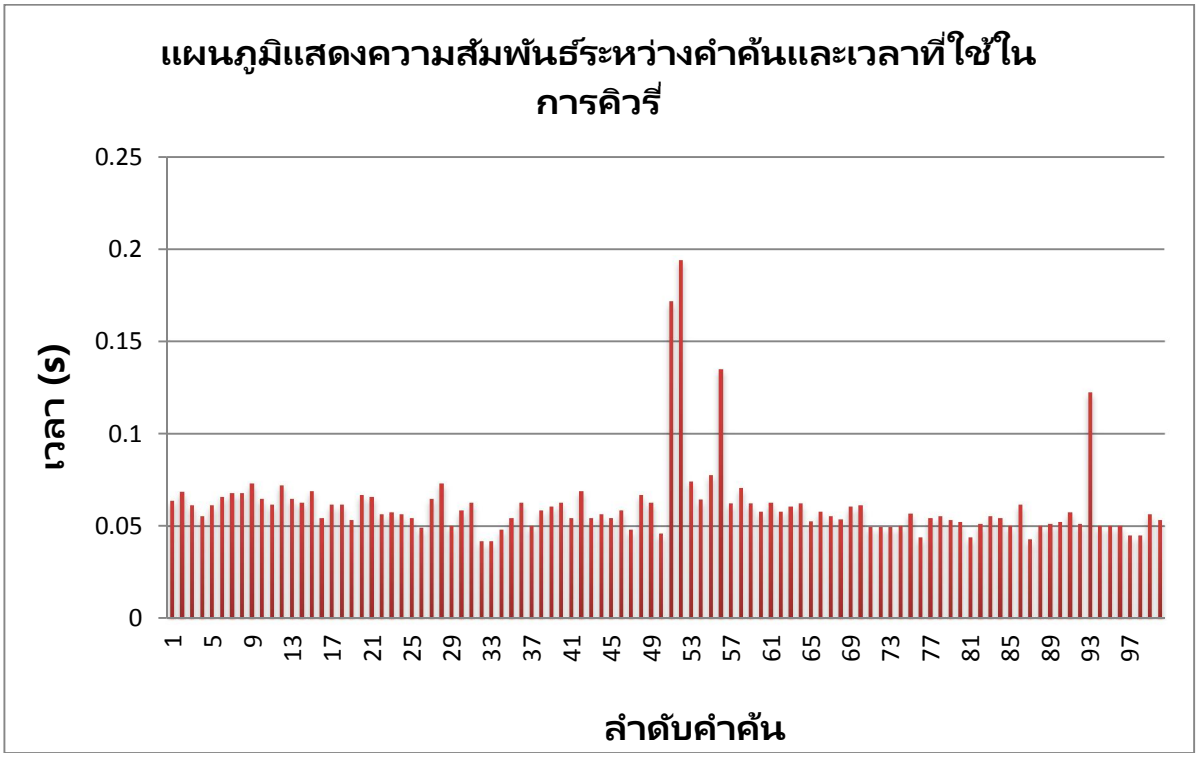
คณะผู้วิจัยทำการทดลองด้วยการสุ่มการค้นหาแบบ Advanced Search ขึ้นมาดังคอลัมน์เงื่อนไขการค้นหาในตารางภาคผนวก ข. ของโครงการ ทั้งนี้เครื่องคอมพิวเตอร์ที่ใช้ในการทดสอบมีคุณสมบัติ Intel Core i5-3230M ความเร็ว 2.60 GHz หน่วยความจำ ขนาด 4 GB DDR3 ฮาร์ดดิสก์ขนาด 640 GB การ์ดอีเทอร์เนตและใช้ระบบปฏิบัติการ Windows 7 เป็นต้น

ผลการทดลองวัดเวลาที่ใช้ในการค้นหาพบว่า แต่ละตัวอย่างที่ใช้ทดลองคิวรี่ใช้เวลาในการค้นหาใกล้เคียงกัน และใช้เวลาไม่มาก โดยเฉลี่ยใช้เวลาน้อยกว่า 1 วินาที (เวลาโดยเฉลี่ย 0.061618 วินาที) ทั้งนี้ความซับซ้อนของคำค้นและจำนวนผลลัพธ์ของการสืบค้นไม่ส่งผลให้ใช้เวลาในการค้นหาแตกต่างกันมาก



ภาพที่ 4.2 แผนภูมิแสดงความสัมพันธ์ระหว่างจำนวนผลลัพธ์และเวลาที่ใช้ในการคิวรี่

ภาพที่ 4.2 แสดงแผนภูมิความสัมพันธ์ระหว่างจำนวนผลลัพธ์และเวลาที่ใช้ในการคิวรี่ แสดงให้เห็นว่า จำนวนผลลัพธ์ของการสืบค้นไม่ส่งผลให้เวลาที่ใช้ในการคิวรี่แตกต่างกันมาก แต่ละตัวอย่างที่ใช้ทดลองในการคิวรี่มีเวลาใกล้เคียงกัน และใช้เวลาโดยเฉลี่ยไม่ถึง 1 วินาที (เวลาเฉลี่ย 0.061618 วินาที) ทั้งนี้จากแผนภูมิจะเห็นได้ว่ากราฟที่ได้มีลักษณะเป็นเส้นตรง และมีบางข้อมูลที่มีค่าเวลาที่ใช้ในการคิวรี่มากกว่าข้อมูลอื่นๆ เนื่องจากมีปัจจัยภายนอกที่ส่งผลให้เวลาที่ใช้ในการคิวรี่มากขึ้น เช่น ความเร็วของอินเทอร์เน็ต เป็นต้น



ภาพที่ 4.3 แผนภูมิแสดงความสัมพันธ์ระหว่างคำค้นและเวลาที่ใช้ในการคิวรี่

ภาพที่ 4.3 แผนภูมิแสดงความสัมพันธ์ระหว่างคำค้นและเวลาที่ใช้ในการคิวรี่ แสดงให้เห็นว่าความซับซ้อนของคำค้นไม่ส่งผลให้เวลาที่ใช้ในการคิวรี่แตกต่างกันมาก แต่ละตัวอย่างที่ใช้ทดลองในการคิวรี่มีเวลาใกล้เคียงกันและใช้เวลาโดยเฉลี่ยไม่ถึง 1 วินาที (เวลาเฉลี่ย 0.061618 วินาที) ทั้งนี้จากแผนภูมิจะเห็นได้ว่าความสูงของกราฟที่ได้มีลักษณะเป็นเส้นตรง และมีบางข้อมูลที่มีค่าเวลาที่ใช้ในการคิวรี่มากกว่าข้อมูลอื่นๆ เนื่องจากมีปัจจัยภายนอกที่ส่งผลให้เวลาที่ใช้ในการคิวรี่มากขึ้น เช่น ความเร็วของอินเทอร์เน็ต เป็นต้น

4.2.2 **การประเมินเวลาของการนำเข้าข้อมูล**

ตารางที่ 4.2 แสดงวลาการค้นหาด้วย Java Concurrency โดยการแตกเทรด โดยทดสอบกับเครื่องคอมพิวเตอร์ที่มี 4 คอร์ Intel ® Core™ i3 CPU M 350 @ 2.27GHz (64-bit) และ RAM 3.7 GB โดยทดสอบกับเว็บไซต์ในตาราง ซึ่งจะเห็นว่าเมื่อมีจำนวนเทรดมากขึ้น จะช่วยให้เวลาที่ใช้ในการคิวรี่ลดลง

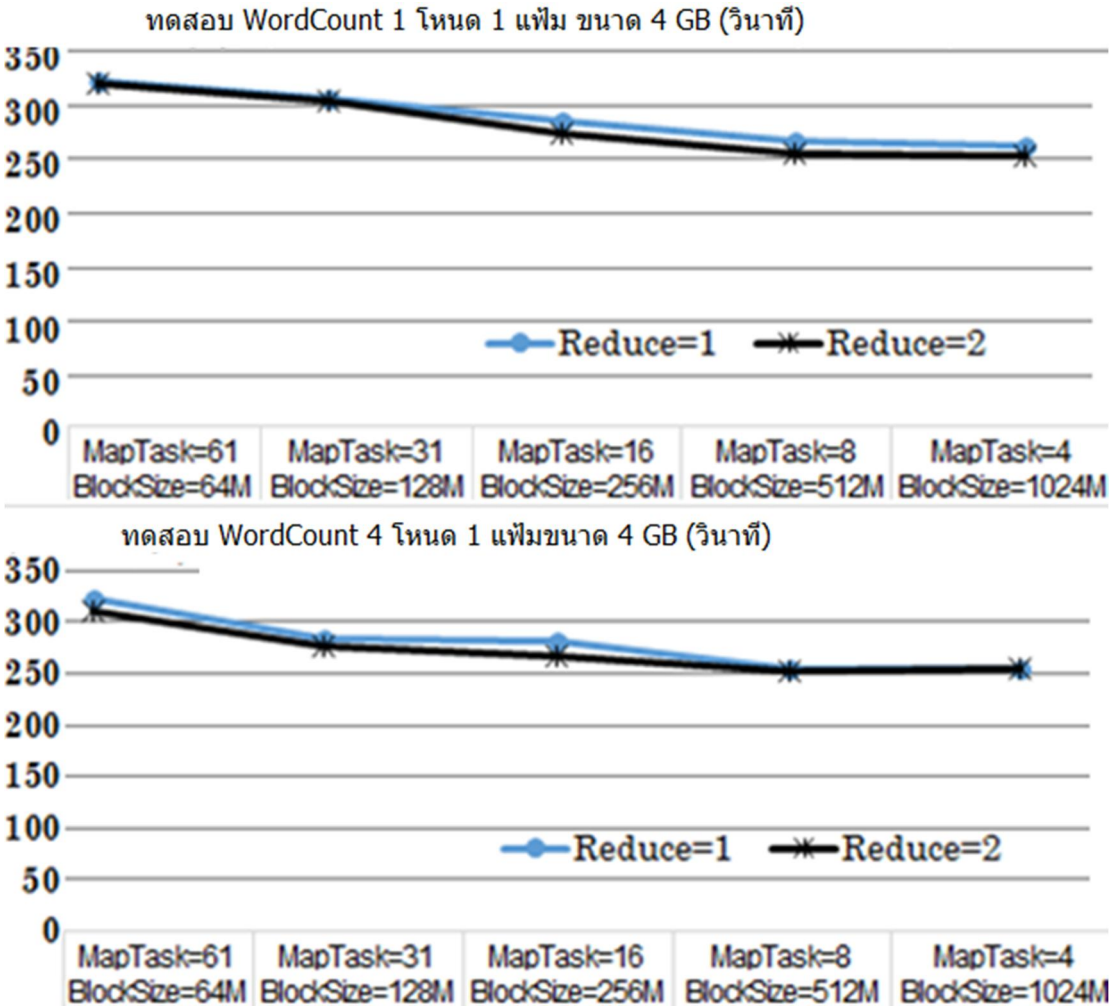
ภาพที่ 4.4 เวลาในการทำงานหลายงานของ MapReduce เมื่อทำการรวมแฟ้มข้อมูล โดยในเบื้องต้นทดสอบกับข้อมูลจาก DBpedia BTC2012[12] โดยแฟ้มข้อมูลมีขนาด 4GB จำนวน 21,398,608 Triples

[12] http://km.aifb.kit.edu/projects/btc-2012/

ตารางที่ 4.2 เวลาการค้นหาด้วย Java Concurrency

Website/Thread	1 เทรด(ms)	2 เทรด (ms)	4 เทรด (ms)
Heyhuahin (82 webs)	4,602	2,529	1,994
Atsiam (146 webs)	100,012	64,728	45,763
Tripadvisor (247 webs)	270,121	135,600	73,981
Agoda (237 webs)	194,443	143,590	74,935

มีการใช้ 3 โหนด Slave และมีโหนดหลัก Master จำนวน 1 โหนด ซึ่งมี ลักษณะดังนี้ โหนด Slave 2 ตัว เป็น Intel ® Core™ 2 Duo CPU E8400 @ 3.00GHz (64-bit), 1.9 GB RAM และอีกโหนด Slave เป็น Intel ® Core™ i5-3330 CPU@ 3.00GHz (64-bit), 7.7 GB RAM และ โหนด Master เป็น Intel ® Core™ i5-3330 CPU@ 3.00GHz (64-bit), 7.7 GB RAM

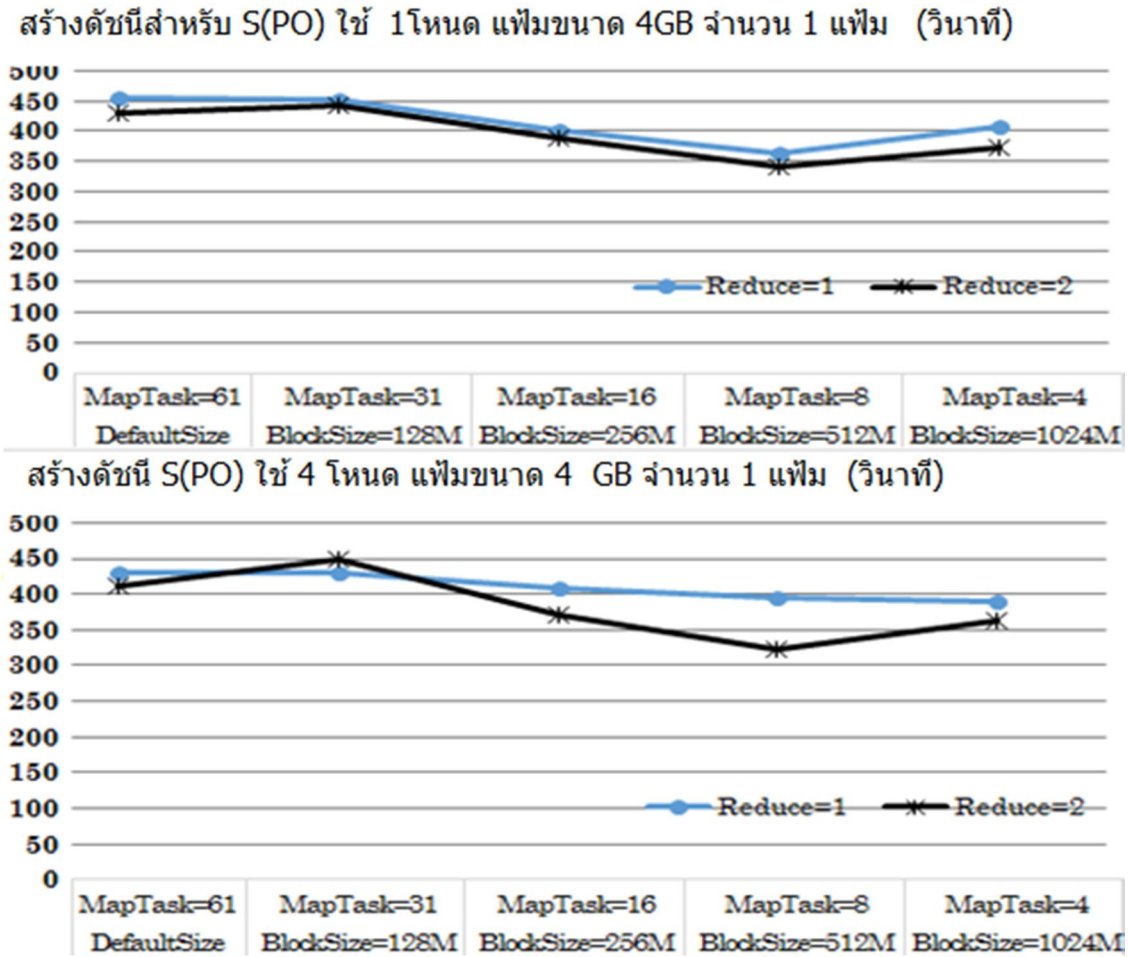


ภาพที่ 4.4 เวลาในการทำงานหลายงานของ MapReduce เมื่อทำการรวมแฟ้มข้อมูลภาพที่ 4.4 ภาพบนคือเวลาของการใช้เพียง 1 โหนด และภาพล่างคือเวลาของการใช้ 4 โหนด โดยงาน Map มีจำนวนตั้งแต่ 4 ถึง

61 และขนาดบล๊อกข้อมูลมีขนาดตั้งแต่ 64, 128, 256, 512 และ 1024 MB และทดสอบกับงาน Reduce 1 หรือ 2 งาน

ผลการทดลองพบว่าเวลาที่ใช้ลดลงเมื่อจำนวนงาน Map ลดลง และเมื่อขนาดบล๊อกข้อมูลใหญ่ขึ้น สำหรับคลัสเตอร์ขนาดเล็กนี้ จากงานวิจัย [1,2] พบว่าจำนวนงาน Map จะต้องสอดคล้องกับจำนวนคอร์ทั้งหมดที่มี และจำนวนงาน Reduce มีเพียง 1 หรือ 2 งาน ก็เพียงพอ แต่ควรกำหนดขนาดบล๊อกข้อมูลให้ใหญ่ให้เหมาะสมเพียงพอเพื่อให้เวลาการทำ I/O ไม่มากเกินกว่าเวลาคำนวณในบล๊อกนั้นจริงๆ

ภาพที่ 4.5 แสดงเวลาในการสร้าง S(PO) ดัชนีในการจัดกลุ่มค่า Predicate และ Object สำหรับแต่ละ Subject ด้วย MapReduce เช่นกัน เป็นการรวมต่ออักขระ โดยใช้ key เดียวกันคือ Subject ซึ่งจะเป็นการรวมข้อมูลในลักษณะเดียวกับภาพ 4.4



ภาพที่ 4.5 เวลาในการสร้าง Index โดยใช้ 1 โหนดและ 4 โหนด

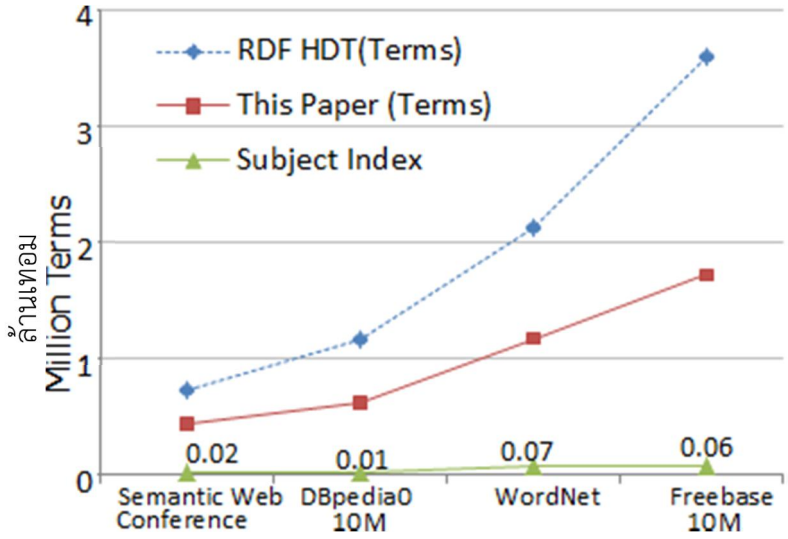
ผลการทดสอบพบว่าประสิทธิภาพด้านเวลาของการใช้ 1 โหนดกับ 4 โหนด ในกรณีนี้พบว่า 4 โหนดทำงานได้เร็วกว่า เมื่อให้จำนวนงาน Map เป็น 8 ถ้ามีจำนวนโหนดมากขึ้น ก็จะเห็นผลด้านเวลาดีขึ้น และเมื่อกำหนดขนาดของแฟ้มให้ใหญ่ในระดับ GB การทำงานของ MapReduce เป็นการทำงานที่เน้นข้อมูลมากๆ และมีการ Overlap การคำนวณระหว่างงานเมื่องาน Map ทำงานเสร็จ Reduce ก็ทำงานต่อได้ และ

งาน Map อื่นๆ ก็ทำงานต่อไป จึงจะทำให้การทำงานนั้นต่อเนื่องงาน Map จึงต้องประมวลผลกับข้อมูลจำนวนมาก นอกจากนี้เวลาในการเริ่มต้นต่างๆ เช่นการเริ่มต้นของ Java Virtual Machine (JVM) ก็ไม่ต่ำกว่า 0.3-0.5 วินาที[13,14] ดังนั้น ขนาดข้อมูลในบล๊อกควรใหญ่พอเพียงด้วยให้งาน Map ทั้งหลายทำงานพร้อมกันด้วย แต่ก็ไม่ทำให้เกิดจำนวนงาน Map มากเกินไปเพราะจะให้ Overhead มากเนื่องจากมีงานในระบบเป็นจำนวนงานเกินไปเทียบกับจำนวนคอร์จริงของเครื่องที่มี

เมื่อเทียบกับเวลาการรวมแฟ้มข้อมูลทั่วไปด้วยโปรแกรม Java แบบลำดับจะใช้เวลามากกว่า 10 นาทีสำหรับข้อมูลขนาด 4GB หรือบางครั้งอาจจะรันไม่ได้เลย การปรับพารามิเตอร์ต่างๆ เช่นจำนวนงาน Map จำนวนงาน Reduce ขนาดของบล๊อก ขนาดของแฟ้มข้อมูลทั้งหมด จำนวน VM และอื่นๆ ของ Hadoop ให้เหมาะกับคลัสเตอร์ที่มีค่อยข้างขึ้นกับลักษณะของงานและคลัสเตอร์อย่างมาก ดังได้กล่าวไว้ใน [1-5]

4.2.3 **การประเมินเวลาของการค้นหาด้วยต้นไม้ไบนารี่**

เมื่อใช้การแทนค่าแบบไบนารี่หรือต้นไม้ 0-1 จะเห็นว่าจะมีขั้นตอนเพิ่มกว่าการใช้รูปแบบ RDF ทั่วไป เพราะจำต้องมีการแปลงจาก RDF เป็น HDT และจาก HDT เป็นไบนารี่ จากนั้นจึงนำข้อมูลที่ผ่านการแปลงเรียบร้อยแล้วเข้าหน่วยประมวลผลกราฟิกส์ เพื่อทำการค้นหา การแปลงเป็นไบนารีย่อมได้เปรียบในการย่อขนาดข้อมูลซึ่งจะทำให้สามารถนำข้อมูลเข้าไปในหน่วยประมวลผลกราฟิกส์ได้มากกว่ารูปแบบ RDF ทั่วไป ดังนั้นในขั้นแรกคณะผู้วิจัยทดสอบวัดขนาดของข้อมูลในรูปแบบไบนารี



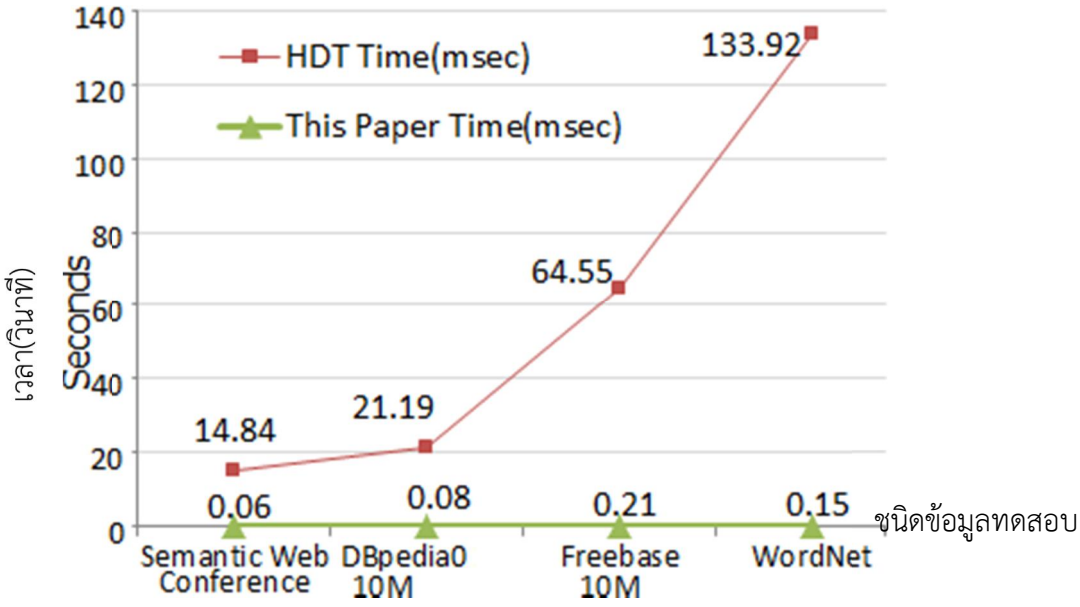
ภาพที่ 4.6 เปรียบเทียบจำนวนเทอม (Term) สำหรับตัวอย่างข้อมูลต่างๆ กันในแกน สำหรับวิธีแทนค่าข้อมูลแบบดั้งเดิม (RDF) แบบ HDT และแบบไบนารี่ในงานวิจัยนี้

---

13 http://stackoverflow.com/questions/10862096/hadoop-map-reduce-how-to-speed-up-job-launch-setup.

14 http://blog.griddynamics.com/2010/07/war-story-optimizing-one-hadoop-job.html.

คณะผู้วิจัยได้ทำการเปรียบเทียบจำนวนเทอมในหน่วยหลักล้านของแฟ้มข้อมูลในรูปแบบ RDF รูปแบบ HDT และรูปแบบไบนารี่ที่นำเสนอ จะเห็นว่าในงานวิจัยนี้นำเสนอรูปแบบข้อมูลที่มีขนาดเล็กกว่าแบบดั้งเดิมและแบบ HDT ในทุกกรณี ดังภาพที่ 4.6



ภาพที่ 4.7 เปรียบเทียบเวลาในการขนย้ายข้อมูลจาก CPU ไปยัง GPU

จากนั้นทำการทดสอบวัดเวลาในการโอนย้ายข้อมูลจากหน่วยประมวลผลกลางไปหน่วยประมวลผลกราฟิกส์ เมื่อข้อมูลเป็นแบบไบนารีจะทำให้สามารถโอนย้ายข้อมูลได้เร็วกว่า ดังภาพ 4.7

ตารางที่ 4.3 และตารางที่ 4.4 แสดงเวลาที่ใช้ในการค้นหาข้อมูลในหน่วยประมวลผลกลาง และหน่วยประมวลผลกราฟิกตามลำดับ เมื่อข้อมูลอยู่ในรูปแบบ HDT ที่ได้กำหนดการสุ่ม ID ของ Subject, Predicate, Object จำนวน 1,024 ตัว และวัดเวลาเฉลี่ยของการค้นหาจำนวน 25 ครั้ง

ตารางที่ 4.5 แสดงเวลาที่ใช้ในการค้นหาข้อมูลในหน่วยประมวลผลกราฟิก เมื่อข้อมูลอยู่ในรูปแบบ SPO โดยใช้ข้อมูลต่อไปนี้ในการทดสอบ swdf, wn3.0, Freebase และ DBPedia ซึ่งแสดงขนาดข้อมูลเป็นจำนวนบิต แสดงจำนวนตำแหน่งของ Subject, Predicate และ Object รวมทั้งจำนวนตำแหน่งทั้งหมดของคำค้นหรือ pattern ที่มี รวมทั้งแสดงข้อมูลเวลาในมิลลิวินาที (ms) และ ข้อมูลของ Kernel ในหน่วยประมวลผลกราฟิกส์ ได้แก่ จำนวนบล็อกต่อกริด และจำนวนเทรดต่อบล็อกจากข้อมูลในตารางแสดงให้เห็นว่าวิธีการแบบไบนารี สามารถนำเข้าข้อมูลได้มากกว่า และค้นหาข้อมูลได้พร้อมกันหลายตำแหน่ง ทำให้ใช้เวลาน้อยกว่า อย่างไรก็ดีการใช้วิธีนี้จำเป็นต้องมีเวลาที่ต้องใช้ในการแปลงข้อมูลมาพิจารณาด้วย ภาพที่ 4.8 แสดงลักษณะข้อมูลแต่ละตัวอย่างทดสอบใน ตารางที่ 4.5

ตารางที่ 4.3 เวลาในการค้นหาข้อมูลด้วย HDT ในหน่วยประมวลผลกลาง

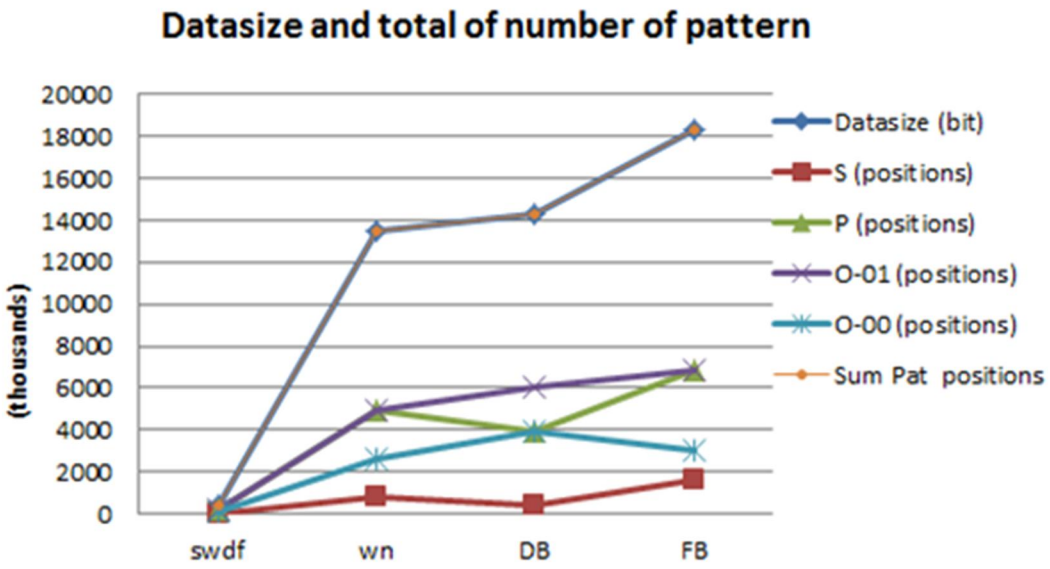
ชนิดข้อมูล	ขนาดข้อมูลต้นฉบับ (MB)	จำนวนเซตของ patterns	เวลา (วินาที)
Freebase	109	1024	54
DBpedia	507	1024	35

ตารางที่ 4.4 เวลาในการค้นหาข้อมูลในรูปแบบ HDT ในหน่วยประมวลผลกราฟิกส์

ชนิดข้อมูล	ขนาดข้อมูล (B)	Patterns	เวลา (วินาที)	จำนวนบล๊อก	จำนวนเทรด
Freebase	1793413	1	7.16	588	1024
DBpedia	1626803	1	6.54	1751	1024

ตารางที่ 4.5 เวลาการค้นหา SPO ในหน่วยประมวลผลกราฟิกส์ (GTX560Ti)

HDT	Data size (bit)	S_11 (positions)	P_10 (positions)	O(00) (positions)	O(01) (positions)	Sum Pat positions	Times (ms)	blocks/ grid	threads block
swdf	433825	23282	168104	168111	73915	410,130	0.87	424	1024
Wn	13,462,923	819,505	4,983,833	4,983,783	2,662,655	13,449,776	26.68	13148	1024
Freebase	14,303,370	386,361	3,932,948	6,037,265	3,932,827	14,289,401	28.32	13969	1024
DBpedia	18,331,369	1,580,553	6,845,230	6,845,180	3,042,504	18,313,467	34.17	17902	1024



ภาพที่ 4.8 เปรียบเทียบขนาดของข้อมูลและตำแหน่ง SPO ที่พบในแต่ละข้อมูล

4.3 การประเมินความพึงพอใจของผู้ใช้บริการเว็บเชิงความหมาย

ในการวิจัยได้ทำการประเมินความพึงพอใจของผู้ใช้งานระบบเว็บเชิงความหมาย กำหนดให้ผู้ใช้งานระบบจำนวน 100 คนทำแบบสอบถามเพื่อประเมินประสิทธิภาพการทำงานของระบบใน 3 ด้านคือ

- ความพึงพอใจด้านคุณภาพของข้อมูล
- ความพึงพอใจด้านการออกแบบและการจัดรูปแบบ
- ความพึงพอใจด้านประโยชน์ของข้อมูลที่ได้รับ

ผลการประเมินความพึงพอใจด้านคุณภาพของข้อมูลเป็นดังตาราง 4.6 จะเห็นได้ว่า ค่าเฉลี่ยของความพึงพอใจด้านคุณภาพของข้อมูลเท่ากับ 3.96 และมีค่าเฉลี่ยของส่วนเบี่ยงเบนมาตรฐานเท่ากับ 0.57 ดังนั้นระบบที่ได้พัฒนาขึ้นมีประสิทธิภาพในด้านคุณภาพของข้อมูลอยู่ในระดับดี

ตารางที่ 4.6 ผลการประเมินความพึงพอใจด้านคุณภาพของข้อมูล

หัวข้อการประเมิน	**ค่าเฉลี่ย (Mean)**	**ส่วนเบี่ยงเบนมาตรฐาน (SD)**	**ความหมาย**
ความรวดเร็วในการเข้าถึงข้อมูล	3.95	0.53	ดี
การเชื่อมโยงข้อมูลภายในเว็บไซต์	4.13	0.65	ดี
ความครบถ้วนสมบูรณ์ของข้อมูลต่างๆ	3.56	0.76	ดี
ความครอบคลุมในการสืบค้นในเชิงกว้างและสามารถสืบค้นข้อมูลแบบเจาะจงได้	4.07	0.66	ดี
ผลลัพธ์ที่ได้ตรงกับความต้องการ	4.07	0.25	ดี
ค่าเฉลี่ย	3.96	0.57	ดี

ผลการประเมินความพึงพอใจด้านการออกแบบและการจัดรูปแบบเป็นดังตาราง 4.7 การประเมินความพึงพอใจด้านการออกแบบและการจัดรูปแบบซึ่งได้ผลลัพธ์ว่าความพึงพอใจด้านนี้มีค่าเฉลี่ยเท่ากับ 4.08 และมีค่าส่วนเบี่ยงเบนมาตรฐานโดยเฉลี่ยเท่ากับ 0.54 ดังนั้นระบบที่ได้พัฒนาขึ้นมีประสิทธิภาพในด้านการออกแบบและการจัดรูปแบบอยู่ในระดับดี

ตารางที่ 4.7 ผลการประเมินความพึงพอใจด้านการออกแบบและการจัดรูปแบบ

หัวข้อการประเมิน	**ค่าเฉลี่ย (Mean)**	**ส่วนเบี่ยงเบนมาตรฐาน (SD)**	**ความหมาย**
ความสวยงามและน่าสนใจของหน้าจอ	3.91	0.58	ดี
การจัดรูปแบบง่ายต่อการอ่านและการใช้งาน	3.88	0.55	ดี
ความง่ายต่อการใช้งานเมนูต่างๆ	4.47	0.56	ดี
ความเหมาะสมของสีสันและขนาดตัวหนังสือบนหน้าจอ	4.08	0.48	ดี
ค่าเฉลี่ย	4.08	0.54	ดี

ผลการประเมินความพึงพอใจด้านประโยชน์ของข้อมูลที่ได้รับในตาราง 4.8 การประเมินความพึงพอใจด้านประโยชน์ของข้อมูลที่ได้รับซึ่งได้ผลลัพธ์ว่าความพึงพอใจด้านนี้มีค่าเฉลี่ยเท่ากับ 3.90 และมีค่าส่วนเบี่ยงเบนมาตรฐานโดยเฉลี่ยเท่ากับ 0.54 ดังนั้นระบบที่ได้พัฒนาขึ้นมีประสิทธิภาพในด้านประโยชน์ของข้อมูลที่ได้รับอยู่ในระดับดี

ตารางที่ 4.8 ผลการประเมินความพึงพอใจด้านประโยชน์ของข้อมูลที่ได้รับ

หัวข้อการประเมิน	**ค่าเฉลี่ย (Mean)**	**ส่วนเบี่ยงเบนมาตรฐาน (SD)**	**ความหมาย**
ให้ข้อมูลสถานที่ให้บริการด้านสุขภาพและสปาที่น่าสนใจ	4.02	0.50	ดี
ให้ข้อมูลเกี่ยวกับสถานที่ใกล้เคียง	3.97	0.55	ดี
แสดงตำแหน่งที่ตั้งและเส้นทางการเดินทาง	4.03	0.57	ดี
ให้ข้อมูลเกี่ยวกับค่าใช้จ่ายโดยประมาณ	3.51	0.50	ดี
ช่วยในการตัดสินใจและการวางแผนการเดินทาง	3.96	0.60	ดี
ค่าเฉลี่ย	3.90	0.54	ดี

ตารางที่ 4.9 ผลการประเมินความพึงพอใจโดยรวม

หัวข้อการประเมิน	**ค่าเฉลี่ย (Mean)**	**ส่วนเบี่ยงเบนมาตรฐาน (SD)**	**ความหมาย**
ความพึงพอใจด้านคุณภาพของข้อมูล	3.97	0.65	ดี
ความพึงพอใจด้านการออกแบบและการจัดรูปแบบ	3.94	0.56	ดี
ความพึงพอใจด้านประโยชน์ของข้อมูลที่ได้รับ	3.97	0.47	ดี
ความพึงพอใจของระบบโดยรวม	3.94	0.44	ดี
ค่าเฉลี่ย	3.96	0.53	ดี

ผลการประเมินความพึงพอใจโดยรวมในตาราง 5.9 ซึ่งได้ผลลัพธ์ว่าความพึงพอใจโดยรวมมีค่าเฉลี่ยเท่ากับ 3.96 และมีค่าส่วนเบี่ยงเบนมาตรฐานโดยเฉลี่ยเท่ากับ 0.53 ดังนั้นระบบที่ได้พัฒนาขึ้นมีประสิทธิภาพโดยภาพรวมอยู่ในระดับดี

บทที่ 5 สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ

ในงานวิจัยนี้ คณะผู้วิจัยได้นำเสนอระบบบริหารจัดการข้อมูลท่องเที่ยวเพื่อสุขภาพ โดยในโครงการย่อยนี้ ได้ทำการสำรวจความต้องการของซอฟต์แวร์ปัจจุบัน รวมทั้งลักษณะโครงสร้างการทำงานของหน่วยงานเทศบาลเมือง พบว่าการค้นหาข้อมูลส่วนใหญ่เป็นการค้นหาจากเว็บไซต์ทั่วไปเช่น Agoda, TripAdvisor ผ่าน Google Search Engine เป็นต้น นอกจากนี้ข้อมูลอื่นๆ มาจากการเดินสำรวจ ซอฟต์แวร์ในการจัดเก็บข้อมูล และเก็บสถิติได้แก่ Excel และ Word เพื่อทำรายงานสรุปต่างๆ แหล่งข้อมูลต่างๆ ในการรวบรวมข้อมูลสถานประกอบการได้แก่ จากเว็บไซต์ Agoda, TripAdvisor, AtSiam, HeyHuaHin และจาก Official Website ของสถานบริการ รวมทั้งการเดินสำรวจในพื้นที่หัวหินและชะอำ ข้อมูลเหล่านี้ถูกนำเข้าออนโทโลยี เชื่อมต่อกับหน้าจอผู้ใช้งานที่ได้ออกแบบไว้ เพื่อสามารถค้นหาได้ สำหรับการประมวลผลการค้นหาเมื่อผู้ใช้ระบุเงื่อนไขการค้นหาหาจะถูกแปลงเป็นคิวรี่ส่งไปยังมิดเดิลแวร์เพื่อติดต่อกับออนโทโลยี และค้นหาข้อมูลที่ต้องการขึ้นมาแสดงผล

คณะผู้วิจัยการนำข้อมูลเข้าแบบขนานด้วยเทคโนโลยี Java แบบพร้อมกัน(Concurrent) และ MapReduce โดยพบว่าการนำเข้าข้อมูลโดยการค้นหาเว็บแบบพร้อมกัน ทำให้ทำงานได้เร็วขึ้น ประมาณ 3-4 เท่าสำหรับกรณีใช้ 4 เทรด และการทำงานแบบ MapReduce จะช่วยทำการรวมข้อมูลจากแฟ้มข้อมูลขนาดใหญ่เช่น 4 GB ได้ดีภายในเวลาไม่เกิน 7 วินาที เทียบกับวิธีปกติโดยโปรแกรมภาษา Java แบบ 1 เทรดที่ใช้เวลาร่วมสิบกว่านาที นอกจากนี้คณะผู้วิจัยได้ทำการวัดประสิทธิภาพการค้นหาจากออนโทโลยี โดยจะพบว่าเวลาเฉลี่ยในการค้นหาไม่เกิน 30 วินาที และทำการนำเสนอโมเดลการค้นหาด้วยวิธีการแบบขนานในโมเดลต้นไม้แบบไบนารีกับหน่วยประมวลผลกราฟิกส์เพื่อประมวลผลแบบขนาน โดยนำข้อมูลออนโทโลยีในรูป Triple เข้ามาในหน่วยประมวลผลกราฟิกส์เพื่อค้นหาแบบพร้อมกัน ซึ่งสามารถนำเข้าข้อมูลได้มากขึ้นและนำมาค้นหาได้เร็วขึ้นนั่นเอง

คณะผู้วิจัยได้ประเมินความพึงพอใจจากผู้ใช้จำนวน 200 คนเพื่อวัดความพึงพอใจของข้อมูลและการออกแบบหน้าเว็บอีกด้วย การออกแบบระบบด้วยออนโทโลยีให้ข้อดี ในการค้นหาข้อมูลต่างๆ ได้ตรงตามความต้องการของผู้ใช้มากขึ้น เนื่องจากเป็นการค้นหาตามความหมายของคำค้นในบริบทที่ผู้ใช้ต้องการจริง อย่างไรก็ดีเครื่องมือที่สนับสนุนการออกแบบออนโทโลยียังมีขีดจำกัดในด้านต่างๆ เช่น กลไกการอนุมานลอจิกที่รองรับ ความเสถียรของระบบเชื่อมต่อ และการทำงานที่ช้ากว่าระบบฐานข้อมูลเชิงสัมพันธ์ที่มีอยู่ ดังนั้นการพัฒนาอย่างต่อเนื่องของเทคโนโลยีเหล่านี้อาจจะทำให้ระบบค้นหาเชิงความหมายสมบูรณ์ขึ้น

นอกจากนี้การรองรับการประมวลผลแบบขนานของเครื่องมือซอฟต์แวร์ที่มีอยู่ยังมีขีดจำกัด ทำให้เมื่อต้องการใช้กลไกการค้นหาแบบขนานมาช่วย ต้องทำการออกแบบการเชื่อมต่อ รวมทั้งเวลาการประมวลผลขั้นก่อนการค้นหาหรือ Preprocess ต่างๆ ค่อนข้างมากซึ่งเป็น Overhead ในการทำงานโดยรวมอยู่ ตาราง 5.1 สรุปกิจกรรมต่างๆ ที่ได้ดำเนินการในโครงการนี้

ตารางที่ 5.1 ตารางเปรียบเทียบวัตถุประสงค์กิจกรรมที่วางแผนไว้และกิจกรรมที่ดำเนินการมาและผลที่ได้รับของ โครงการย่อย 1

กิจกรรมที่ได้วางแผนไว้	ผลที่ได้รับ
1. สำรวจความต้องการซอฟต์แวร์ในปัจจุบัน	รายงานผลสำรวจความต้องการซอฟต์แวร์ จากผู้ใช้งานในหน่วยงาน (บทที่ 4 หัวข้อ 4.1) และภาคผนวก ก ค ง และ จ ของแผนงาน
2. การศึกษางานวิจัยที่เกี่ยวข้องด้านการประมวลแบบขนานบนกลุ่มเมฆ	รายงานงานวิจัยที่เกี่ยวข้อง (บทที่ 2)
3. พัฒนาตัวแบบของระบบการประมวลบนกลุ่มเมฆ	ตัวแบบการประมวลบนกลุ่มเมฆ (บทที่ 3 หัวข้อ 3.5)
4. พัฒนาอัลกอริทึมการประมวลคิวรี่แบบขนาน และการแทนค่าข้อมูล	อัลกอริทึมแบบขนาน สำหรับการประมวลคิวรี่ รูปแบบการแทนค่าข้อมูล (บทที่ 3 หัวข้อ 3.1, 3.2 และภาคผนวก ก)
5. พัฒนาโปรแกรมการประมวลผลคิวรี่แบบขนาน	โปรแกรมการประมวลคิวรี่แบบขนาน
6. การออกแบบหน้าจอ การติดต่อผู้ใช้ ระบบออนโทโลยี	แบบหน้าจอ การติดต่อผู้ใช้ (บทที่ 3 หัวข้อ 3.3)
7. ออกแบบฐานข้อมูลระบบ ได้แก่ ข้อมูลผู้ใช้ระบบ ข้อมูลสถานที่ท่องเที่ยว พัฒนาระบบการค้นหา พัฒนาโปรแกรมทดลองใช้	ฐานข้อมูลผู้ใช้ระบบและข้อมูลเกี่ยวกับการท่องเที่ยว ระบบค้นหา และการจัดการ ในรายงานแผนงาน ภาคผนวก ช. และคู่มือนำเข้าข้อมูลในรายงานโครงการย่อย 1 ภาคผนวก ค
8. เปรียบเทียบผลการค้นหาคิวรี่ ด้านประสิทธิภาพเวลา	ผลการประเมินด้านประสิทธิภาพ (บทที่ 4 หัวข้อ 4.2 และภาคผนวก ข)
9. ประเมินผลการใช้งานกับผู้ใช้บทบาทต่างๆ กัน	ผลการประเมินผู้ใช้งาน (บทที่ 4 หัวข้อ 4.3)

กิจกรรมที่ได้วางแผนไว้	ผลที่ได้รับ
10. ประเมินค่าใช้จ่ายถ้าจะนำระบบไปใช้จริง	ประมาณการค่าใช้จ่าย ในรายงานแผน ภาคผนวก ฉ. และ ช.
11. จัดเตรียมการอบรมผู้ใช้ จัดทำคู่มือผู้ใช้งาน	การอบรมผู้ใช้งาน คู่มือการใช้งาน ในรายงานแผน ภาคผนวก ญ.
12. เขียนสรุปรายงานและเอกสารวิชาการ	สรุปรายงานการวิจัย และเอกสารวิชาการใน ภาคผนวก ฌ.

5.1 ข้อเสนอแนะ

ในงานวิจัยในอนาคต การพัฒนาโครงสร้างข้อมูลการแทนค่าออนโทโลยีพื้นฐานนี้ให้เหมาะกับการประมวลผลแบบขนานไม่ว่าจะด้วยลักษณะคลัสเตอร์ หรือหน่วยประมวลผลกราฟิกส์ก็ตาม ไลบารีต่างๆ ที่รองรับจะมีส่วนสำคัญในการเชื่อมต่อการประมวลผลออนโทโลยีกับเว็บเชิงความหมายให้เป็นอัตโนมัติ และด้วยประสิทธิภาพที่ดีต่อไป เมื่อพัฒนาการทางฮาร์ดแวร์ของหน่วยประมวลแบบหลายแกนดีมากขึ้น มีราคาถูกลง การซอฟต์แวร์รองรับการพัฒนาโปรแกรมแบบหลายประมวลหลายแกนมามากขึ้น รวมทั้งชุดพัฒนาต่างๆ จะทำให้การพัฒนาการค้นหาข้อมูลเชิงความหมายเป็นไปได้ง่ายมากขึ้น

อย่างไรก็ตามองค์ความรู้ในการพัฒนางานประยุกต์ที่ได้นี้ ก็จะมีส่วนสำคัญในการสร้างนักวิจัยแบบทางคอมพิวเตอร์ในการพัฒนางานวิจัยที่รองรับการทำงานแบบพร้อมกันหรือบนสถาปัตยกรรมแบบหลายแกนต่อไปในอนาคตต่อไป

เอกสารอ้างอิง

[1] Bernstein, D., and Vij,D., "Using semantic web ontology for Inter cloud directories and exchanges," In *Proceedings of ICOMP'10, the 11th International Conference on Internet Computing,* 2010.

[2] Benedikt, M., and Teuteberg, F., "Risk and compliance management for cloud Computing services: designing a reference model," in *Proceedings of AMCIS,* 2011.

[3] Ming, X. Z., "A semantic grid oriented to etourism", in *Proceedings of the 1st International Conference on Cloud Computing*. 2009.

[4] Lee, C.G., "Health care and tourism: evidence from Singapore," *Tourism Management*, Vol. 31, No. 4, pp. 486-488, 2010.

[5] Prantner, K., Ding, Y., Luger, M., Yan, Z., and Herzog, C., "Tourism ontology and semantic management system: State-of-the-arts Analysis", In: IADIS International Conference WWW/Internet 2007. Vila Real, Portugal, 2007.

[6] McCreadie, R., Mcdonald, C., and Ounis, I., "Comparing distributed indexing: to MapReduce or not?," in *Proceedings of the 7th Workshop on Large-Scale Distributed Systems for Information Retrieval*, 2009.

[7] Lin, J., and Dyer, C., "Data-Intensive Text Processing with MapReduce" , pre-production manuscript of a book in the Morgan & Claypool Synthesis Lectures on Human Language Technologies. Prepared April 3, 2012.

[8] Wei, Z. and JaJa, J. , "A fast algorithm for constructing inverted files on heterogeneous platforms," Journal of Parallel and Distributed Computing, vol. 72, no. 5, pp. 728-738, 2011.

[9] Ming, L., et al., "Authorized private keyword search over encrypted data in cloud computing," in *Proceedings of the 31st International Conference on Distributed Computing Systems, IEEE Computer Society*, 2011.

[10] Kyong-Ha, L., et al., "Parallel data processing with MapReduce: a survey," *SIGMOD Rec.*, Vol. 40, no. 4, pp. 11-20, 2011.

[11] Lin, J., Metzler, D., Elsayed, T., Wang, L., "Of ivory and smurfs: Loxodontan MapReduce experiments for web search," in *Proceedings of the Eighteenth Text REtrieval Conference, TREC*, 2009.

[12] Williams, G. T., Weaver, J., et al., "Scalable reduction of large datasets to interesting subsets," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 8, no. 4, pp. 365-373, 2010.

[13] Urbani, J., et al., "WebPIE: A Web-scale parallel inference engine using MapReduce," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 10, pp. 59-75, 2012.

[14] Mohammad, F.H., McGlothlin, J., Masud, M. M., Khan, L.R., and Thuraisingham, B., "Heuristics-based query processing for large RDF graphs using cloud computing," IEEE Transactions on Knowledge and Data Engineering, vol. 23, no. 9, pp. 1312-1327, Sept. 2011, doi:10.1109/TKDE.2011.103.

[15] Rohloff, K., and Richard, E.S., "High-performance, massively scalable distributed systems using the MapReduce software framework: the SHARD triple-store," in Programming Support Innovations for Emerging Distributed Applications, ACM: Reno, Nevada, 2010.

[16] Hartley, T.D.R., Saule, E., and Çatalyürek, Üm.V., "Improving performance of adaptive component-based dataflow middleware," *Parallel Computing*, Vol. 38, no. 6-7, pp. 289-309, 2012.

[17] Soma, R., and Prasanna, V.K., "Parallel inferencing for OWL knowledge bases", in *Proceedings of the 2008 37th International Conference on Parallel Processing*, IEEE Computer Society, Washington DC, USA, 2008.

[18] Nicholas, C. and V.L. Narasimhan, "A Novel Data Distribution Technique for Host-Client Type Parallel Applications," *IEEE Trans. Parallel Distrib. Syst.*, Vol. 13, No. 2, pp. 97-110, 2002.

[19] Yang, H.-c., Dasdan, A., Hsiao, R.-L., and Parker, D. S, "Map-Reduce-Merge: simplified relational data processingon large clusters," in *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pp. 1029-1040, 2007.

[20] Hiemstra, D. and Hauff, C., "MapReduce for Experimental Search," in *Proceedings of the Nineteenth Text REtrieval Conference (TREC 2010)*, 2010.

[21] Dean, J., and Ghemawat, S., "MapReduce: simplified data processing on large clusters"*,* *Communications of the ACM*, Vol. 50, No. 1, pp. 137-149, 2008.

[22] Fukunaga, A., Kishimoto, A., Botea, A., "Iterative Resource Allocation for Memory Intensive Parallel Search Algorithms on Clouds", *Grids, and Shared Clusters*, 2012.

[23] Rohloff, K., and Schantz, R. E. “Clause-iteration with MapReduce to scalably query datagraphs in the SHARD graph-store,” In *Proceedings of the fourth international workshop on Data-intensive distributed computing (DIDC '11)*, pp. 35-44, 2011. DOI=10.1145/1996014.1996021.

[24] Lin, J., Metzler, D., Elsayed, T., and Wang, L., “Of ivory and smurfs: Loxodontan MapReduce experiments for web search,” In *Proceedings of the Eighteenth Text REtrieval Conference (TREC 2009)*, 2009.

[25] Byun, C., Arcand, W., Bestor, D., Bergeron, D., Hubbell, M., Kepner, J., and Yee, C., “Driving big data with big compute,” *[Online]. Available:* http://www.mit.edu/~kepner/pubs/ByunKepner_2012_BigData_Paper.pdf.

[26] Rohloff, K., Cloud computing for Scalability:The SHARD Triple-Store, [Online]. Available: https://dist-systems.bbn.com/people/krohloff/papers/2011/Rohloff_Meetup_01_09_2011.pdf

[27] Apache Hadoop คืออะไร. [Online]. Available: http://www.wegethosting.com/blog/Apache-hadoop-คืออะไร)

[28] MacLean, D., “A very brief introduction to MapReduce.”, Lecture note, http://hci.stanford.edu/courses/cs448g/a2/files/map_reduce_tutorial.pdf.

[29] Jain, S., “Hadoop architecture-types of Hadoop nodes in cluster-Part1,” [Online]. Available: http://www.haaram.com/Compleetearticle.aspx?aid=505847&ln=)

[30] Jain, S., “Hadoop Architecture-Types of Hadoop Nodes in Cluster-Part2,” [Online]. Available: http://www.haaram.com/Compleetearticle.aspx?aid=505847&ln=

[31] Vaidya, M., “Parallel Processing of cluster by Map Reduce,” *International Journal of Distributed and Parallel Systems (IJDPS)*, vol. 3, no. 1, pp. 167-197, 2012.

[32] Lea, D., “A Java Fork/Join framework.” State University of New York at Oswego, Proceedings of ACM 2000 conference on Java Grande (JAVA’00), pp. 36-43, 2000.

[33] FernáNdez, J. D., MartíNez-Prieto, M. A., GutiéRrez, C., Polleres, A., and Arias, M., “Binary RDF representation for publication and exchange (HDT).” *Web Semant,* vol. 19, no. 22–41, 2013.

ภาคผนวก โครงการย่อย 1

ภาคผนวก ก. การติดตั้ง Hadoop และการทดสอบประสิทธิภาพ MapReduce

ภาคผนวก ข. คิวรี่ที่ใช้ทดสอบประสิทธิภาพด้านเวลา

การติดตั้ง hadoop บนระบบปฏิบติการลีนุกซ์ Ubuntu แบบ 1 โหนด

1. **ติดตั้ง java jdk 7 ลงในเครื่อง**

1.1 ดาวน์โหลดจาวาที่มีสกุลไฟล์เป็น .tar.gz* จากนั้นทำการแตกไฟล์ผ่าน terminal โดยจะต้องเข้าไปอยู่ในไดเรกทอรี่ที่เก็บไฟล์จาวาไว้แล้วใช้คำสั่ง

> *cd Downloads* (เข้าไปในไดเรกทอรี่ที่เก็บไฟล์จาวา)

> *tar -xvf jdk-7u45**-linux-i586.tar.gz* (32 bit)

> *tar -xvf jdk-7u45-linux-x64.tar.gz* (64 bit)

* ดาวน์โหลดไฟล์จาวาได้จาก http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html

** 7u45 คือ เวอร์ชันของจาวา ซึ่งจะมีเวอร์ชันใหม่กว่าอีก

1.2 เมื่อแตกไฟล์แล้วจะได้โฟลเดอร์ jdk1.7.0_45 ให้สร้างโฟลเดอร์เพื่อเก็บจาวาโดยเฉพาะ

> *mkdir -p /usr/lib/jvm* (สร้างโฟลเดอร์เองจะไว้ที่ใดก็ได้)

> *sudo mv /home/user/Downloads/jdk1.7.0_45 /usr/lib/jvm/jdk1.7.0* (ย้ายไปโฟลเดอร์ที่สร้างไว้)

รูปแบบการใช้คำสั่งย้ายโฟลเดอร์ คือ *sudo mv [ไดเรกทอรี่ต้นทางมีไฟล์ที่แตกแล้ว] [ไดเรกทอรี่ปลายทาง*]

1.3 กำหนดลำดับความสำคัญของ JDK โดยรันคำสั่งต่อไปนี้

> *sudo update-alternatives --install "/usr/bin/java" "java" "/usr/lib/jvm/jdk1.7.0/bin/java" 1*

> *sudo update-alternatives --install "/usr/bin/javac" "javac" "/usr/lib/jvm/jdk1.7.0/bin/javac" 1*

> *sudo update-alternatives --install "/usr/bin/javaws" "javaws" "/usr/lib/jvm/jdk1.7.0/bin/javaws" 1*

1.4 แก้ไขความเป็นเจ้าของไฟล์และสิทธิ์ของการปฏิบัติงาน โดยรันคำสั่งต่อไปนี้

> *sudo chmod a+x /usr/bin/java*

> *sudo chmod a+x /usr/bin/javac*

> *sudo chmod a+x /usr/bin/javaws*

> *sudo chown -R user:user /usr/lib/jvm/jdk1.7.0*

* java JDK มี executables อื่นๆ อีกมากมายที่สามารถติดตั้งในทำนองเดียวกันกับข้างต้น ซึ่ง java, javac, javaws อาจจะจำเป็นต้องใช้บ่อยที่สุด

```
hduser@hduser-System-Product-Name:~/Downloads$ cd
hduser@hduser-System-Product-Name:~$ cd Downloads
hduser@hduser-System-Product-Name:~/Downloads$ sudo mv jdk1.7.0_45 /usr/lib/jvm/
jdk1.7.0/
hduser@hduser-System-Product-Name:~/Downloads$ cd
hduser@hduser-System-Product-Name:~$ sudo update-alternatives --install "/usr/bin/java" "java"
"/usr/lib/jvm/jdk1.7.0/jdk1.7.0_45/bin/java" 1
update-alternatives: using /usr/lib/jvm/jdk1.7.0/jdk1.7.0_45/bin/java to provide /usr/bin/java
(java) in auto mode.
hduser@hduser-System-Product-Name:~$ sudo update-alternatives --install "/usr/bin/javac" "javac"
"/usr/lib/jvm/jdk1.7.0/jdk1.7.0_45/bin/javac" 1
update-alternatives: using /usr/lib/jvm/jdk1.7.0/jdk1.7.0_45/bin/javac to provide /usr/bin/javac
(javac) in auto mode.
hduser@hduser-System-Product-Name:~$ sudo update-alternatives --install "/usr/bin/javaws" "javaws
" "/usr/lib/jvm/jdk1.7.0/jdk1.7.0_45/bin/javaws" 1
update-alternatives: using /usr/lib/jvm/jdk1.7.0/jdk1.7.0_45/bin/javaws to provide /usr/bin/javaw
s (javaws) in auto mode.
hduser@hduser-System-Product-Name:~$ mkdir ~/.mozilla/plugins/
hduser@hduser-System-Product-Name:~$ ln -s /usr/lib/jvm/jdk1.7.0/jdk1.7.0_45/jre/lib/amd64/libnpjp2.so
~/.mozilla/plugins/
hduser@hduser-System-Product-Name:~$ java -version
java version "1.7.0_45"
Java(TM) SE Runtime Environment (build 1.7.0_45-b18)
Java HotSpot(TM) 64-Bit Server VM (build 24.45-b08, mixed mode)
hduser@hduser-System-Product-Name:~$
```

1.5 จากนั้นรันคำสั่ง

> *sudo update-alternatives --config java*

คำสั่งนี้จะปรากฏลำดับเลข 0, 1, 2 และพาธให้เลือก โดยเลือกเลขที่เป็นพาธของ /usr/lib/jvm/jdk1.7.0/bin/java (พิมพ์ตัวเลขลงใน terminal)

> *sudo update-alternatives --config javac*

เลือกเลขที่เป็นพาธของ /usr/lib/jvm/jdk1.7.0/bin/javac

> *sudo update-alternatives --config javaws*

เลือกเลขที่เป็นพาธของ /usr/lib/jvm/jdk1.7.0/bin/javaws

1.6 ตรวจสอบการติดตั้งจาวาด้วยคำสั่ง

> *java -version*

*** ขั้นตอนการติดตั้งจาวาอ้างอิงจาก http://askubuntu.com/questions/56104/how-can-i-install-sun-oracles-proprietary-java-jdk-6-7-8-or-jre

2. Configuing SSH

Hadoop ร้องขอ SSH เพื่อใช้ในการเข้าถึงโหนดต่างๆ เช่น การทำงานแบบ remote จากเครื่องแม่ (master) ไปเครื่องลูก (slave)

2.1 ติดตั้ง SSH

> *sudo apt-get install openssh-server*

2.2 สร้างคู่คีย์ RSA ด้วยรหัสผ่านที่ว่างเปล่า เพราะไม่ต้องป้อนรหัสผ่านทุกครั้งที่ทำการโต้ตอบกับแต่ละโหนด

> ssh-keygen -t rsa -P "" (จากนั้นกด Enter)

> cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys

3. Disable IPv6

ปิดการใช้งานไอพีรุ่น 6 เพราะ ubuntu ใช้ IP 0.0.0.0 สำหรับการกำหนดค่าที่แตกต่างกันของ Hadoop

3.1 เปิดไฟล์ sysctl.conf ด้วย Text Editor

> sudo gedit /etc/sysctl.conf

3.2 เติมข้อความต่อไปนี้ต่อท้ายไฟล์

#disable ipv6

net.ipv6.conf.all.disable_ipv6 = 1

net.ipv6.conf.default.disable_ipv6 = 1

net.ipv6.conf.lo.disable_ipv6 = 1

export HADOOP_OPTS=-Djava.net.preferIPv4Stack=true

จากนั้นบันทึกและปิดไฟล์ หากติดปัญหา permission ให้ใช้คำสั่งเดียวกันนี้ใน root account

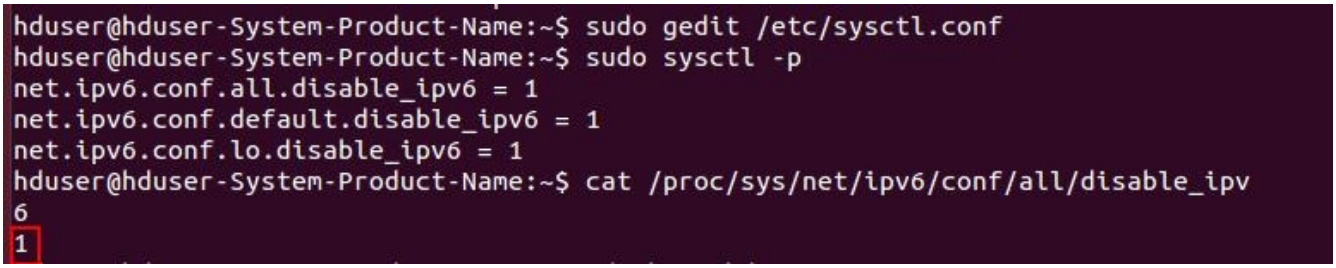
3.3 กำหนดการตั้งค่าเริ่มต้นอีกครั้ง โดยใช้คำสั่ง

> sudo sysctl -p

3.4 ตรวจสอบการปิด IPv6 ด้วยคำสั่ง

> cat /proc/sys/net/ipv6/conf/all/disable_ipv6

หากได้ค่า 1 แสดงว่าทำการปิด IPv6 แล้ว



4. ติดตั้ง Hadoop

4.1 ดาวน์โหลด Hadoop ที่มีสกุลไฟล์ .tar.gz ในคู่มือนี้ดาวน์โหลดเวอร์ชัน 0.20.2

4.2 แตกไฟล์โดยเข้าไปในไดเรกทอรี่ที่เก็บไฟล์ hadoop ไว้ก่อน ในคู่มือนี้เก็บไว้ใน /home

> *sudo tar xzf hadoop-0.20.2.tar.gz*

4.3 เมื่อแตกไฟล์แล้วจะได้โฟลเดอร์ hadoop-0.20.2

> *sudo mv hadoop-0.20.2 hadoop* (เปลี่ยนชื่อโฟลเดอร์เป็น hadoop)

4.4 Update $HOME/.bashrc

> *sudo gedit /home/hduser*/.bashrc* (เปิดไฟล์, *hduser คือ ชื่อผู้ใช้ที่ติดตั้ง hadoop)

4.5 เติมข้อความต่อไปนี้ต่อท้ายไฟล์

```
# Set Hadoop-related environment variables
export HADOOP_HOME=/home/hduser/hadoop*

# Set JAVA_HOME (we will also configure JAVA_HOME directly for Hadoop later on)
export JAVA_HOME=/usr/lib/jvm/jdk1.7.0**
# Some convenient aliases and functions for running Hadoop-related commands
unalias fs &> /dev/null
alias fs="hadoop fs"
unalias hls &> /dev/null
alias hls="fs -ls"

# If you have LZO compression enabled in your Hadoop cluster and
# compress job outputs with LZOP (not covered in this tutorial):
# Conveniently inspect an LZOP compressed file from the command
# line; run via:
#
# $ lzohead /hdfs/path/to/lzop/compressed/file.lzo
#
# Requires installed 'lzop' command.
#
lzohead () {
    hadoop fs -cat $1 | lzop -dc | head -1000 | less
}
```

 # Add Hadoop bin/ directory to PATH

export PATH=$PATH:$HADOOP_HOME/bin

---

* ไดเรกทอรี่ที่เก็บโฟลเดอร์ในข้อ 4.3

** JAVA_HOME จากข้อ 1.2

4.6 แก้ไข JAVA_HOME

> *sudo gedit /home/hduser/hadoop/conf/hadoop-env.sh* (เปิดไฟล์)

แก้ไข **# export JAVA_HOME=/usr/lib/j2sdk1.5-sun** เป็น

export JAVA_HOME=/usr/lib/jvm/jdk1.7.0 (ลบเครื่องหมาย # แล้วเปลี่ยนเป็นไดเรกทอรี่ที่เก็บจาวา)

4.7 สร้างโฟลเดอร์สำหรับเก็บไฟล์ข้อมูลและการทำงานต่างๆ ของ hadoop

> *sudo mkdir /home/hduser/tmp* (สร้างชื่อใดก็ได้ ในที่นี้สร้าง tmp เก็บไว้ที่ /home)

> *sudo chown hduser:hduser /home/hduser/tmp* (แก้ไขสิทธิ์)

> *sudo chmod 750 /home/hduser/tmp* (เพิ่มความปลอดภัย)

4.8 แก้ไขไฟล์ **core-site.xml**

> *sudo gedit /home/hduser/hadoop/conf/core-site.xml* (เปิดไฟล์)

แล้วเติมข้อความต่อไปนี้ระหว่าง **<configuration> .. </configuration>**

```
<!-- In: conf/core-site.xml -->
<property>
  <name>hadoop.tmp.dir</name>
  <value>/home/hduser/tmp*</value>
  <description>A base for other temporary directories.</description>
</property>

<property>
  <name>fs.default.name</name>
  <value>hdfs://localhost:54310**</value>
  <description>The name of the default file system. A URI whose
  scheme and authority determine the FileSystem implementation. The
  uri's scheme determines the config property (fs.SCHEME.impl) naming
```

the FileSystem implementation class. The uri's authority is used to

determine the host, port, etc. for a filesystem.**</description>**

</property>

---

* ไดเรกทอรี่ในข้อ 4.7

** กำหนด file system ในที่นี้ทำในเครื่องตัวเอง (1 โหนด)

4.9 แก้ไขไฟล์ **mapred-site.xml**

> *sudo gedit /home/hduser/hadoop/conf/mapred-site.xml* (เปิดไฟล์)

แล้วเติมข้อความต่อไปนี้ระหว่าง **<configuration> .. </configuration>**

<!-- In: conf/mapred-site.xml -->

<property>

<name>mapred.job.tracker</name>

<value>localhost:54311</value>

<description>The host and port that the MapReduce job tracker runs

at. If "local", then jobs are run in-process as a single map

and reduce task.

</description>

</property>

4.10 แก้ไขไฟล์ **hdfs-site.xml**

> *sudo gedit /home/hduser/hadoop/conf/hdfs-site.xml* (เปิดไฟล์)

แล้วเติมข้อความต่อไปนี้ระหว่าง **<configuration> .. </configuration>**

<!-- In: conf/hdfs-site.xml -->

<property>

<name>dfs.replication</name>

<value>1*</value> * จำนวนโหนด (DataNode)

<description>Default block replication.

The actual number of replications can be specified when the file is created.

The default is used if replication is not specified in create time.

</description>

</property>

---

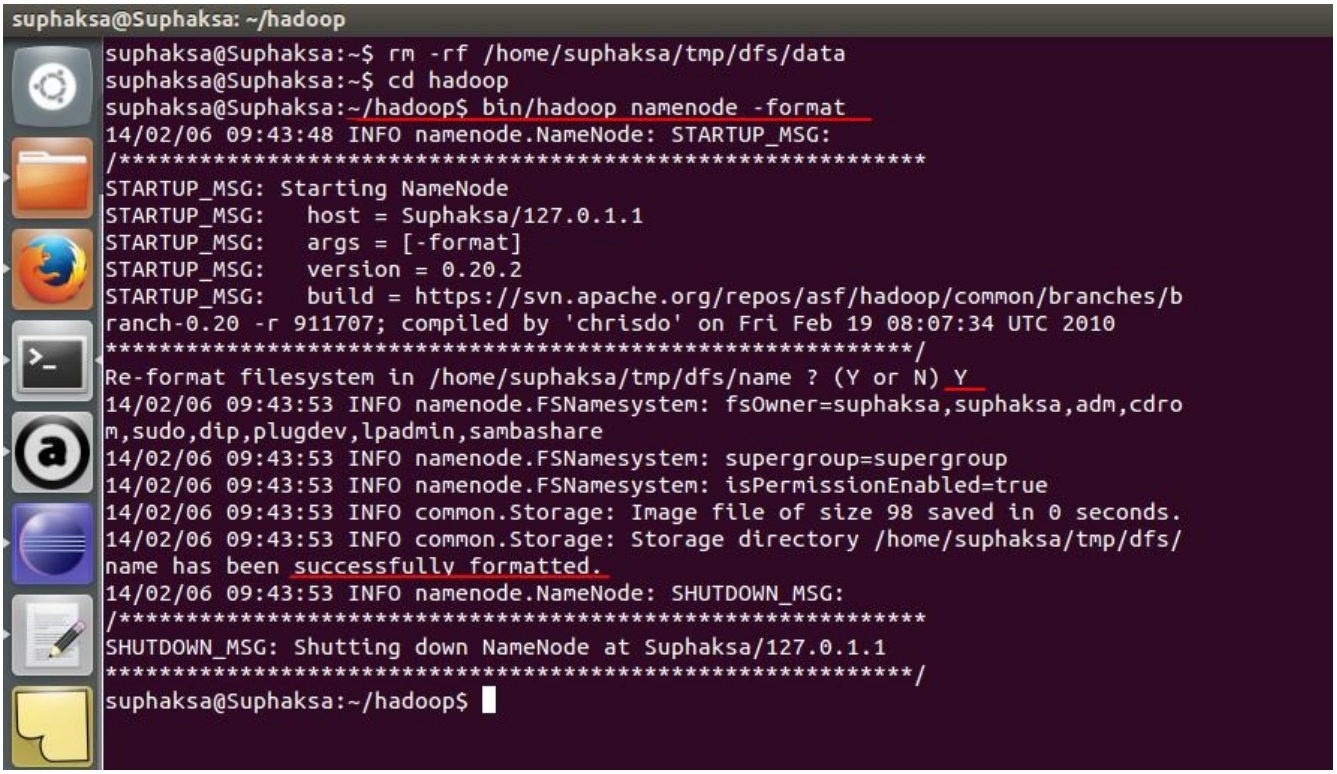
5. Running Hadoop

5.1 ฟอร์แมตเนมโหนดโดยใช้คำสั่ง

> *cd Hadoop* (เข้าไปใน HADOOP_HOME)

> *bin/hadoop namenode -format*

หากปรากฏคำถามว่าจะ re-format filesystem หรือไม่ ให้ตอบ Y



สังเกตการสำเร็จด้วยข้อความ successfully formatted

5.2 สตาร์ตงานทุกประเภทด้วยคำสั่ง

> *bin/start-all.sh*

สังเกตความสำเร็จโดยใช้คำสั่ง

> jps

ปรากฏงานที่สตาร์ตแล้ว ประกอบด้วย Namenode, Datanode, Jobtracker, Tasktracker และ Secondarynamenode

กรณีเกิดปัญหาสตาร์ต DataNode ไม่ได้ ให้แก้ไขโดยการสตาร์ตทีละงานเอง โดยมีขั้นตอนดังนี้

> rm -rf /home/hduser/tmp/dfs/data

> cd Hadoop

> bin/hadoop namenode -format

> bin/hadoop-daemon.sh start namenode

> bin/hadoop-daemon.sh start datanode

> bin/hadoop-daemon.sh start secondarynamenode

> bin/hadoop-daemon.sh start tasktracker

> bin/hadoop-daemon.sh start jobtracker

```
hduser@hduser-System-Product-Name:~$ rm -rf /home/hduser/tmp/dfs/data
hduser@hduser-System-Product-Name:~$ cd hadoop
hduser@hduser-System-Product-Name:~/hadoop$ bin/hadoop namenode -format
13/12/06 08:25:36 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = hduser-System-Product-Name/127.0.1.1
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 0.20.2
STARTUP_MSG:   build = https://svn.apache.org/repos/asf/hadoop/common/branches/b
ranch-0.20 -r 911707; compiled by 'chrisdo' on Fri Feb 19 08:07:34 UTC 2010
************************************************************/
Re-format filesystem in /home/hduser/tmp/dfs/name ? (Y or N) Y
13/12/06 08:25:39 INFO namenode.FSNamesystem: fsOwner=hduser,hduser,adm,cdrom,su
do,dip,plugdev,lpadmin,sambashare
13/12/06 08:25:39 INFO namenode.FSNamesystem: supergroup=supergroup
13/12/06 08:25:39 INFO namenode.FSNamesystem: isPermissionEnabled=true
13/12/06 08:25:39 INFO common.Storage: Image file of size 96 saved in 0 seconds.
13/12/06 08:25:39 INFO common.Storage: Storage directory /home/hduser/tmp/dfs/na
me has been successfully formatted.
13/12/06 08:25:39 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at hduser-System-Product-Name/127.0.1.1
************************************************************/
hduser@hduser-System-Product-Name:~/hadoop$ bin/hadoop-daemon.sh start namenode
starting namenode, logging to /home/hduser/hadoop/bin/../logs/hadoop-hduser-name
node-hduser-System-Product-Name.out
hduser@hduser-System-Product-Name:~/hadoop$ bin/hadoop-daemon.sh start jobtracke
r
starting jobtracker, logging to /home/hduser/hadoop/bin/../logs/hadoop-hduser-jo
btracker-hduser-System-Product-Name.out
hduser@hduser-System-Product-Name:~/hadoop$ bin/hadoop-daemon.sh start tasktrack
er
starting tasktracker, logging to /home/hduser/hadoop/bin/../logs/hadoop-hduser-t
asktracker-hduser-System-Product-Name.out
hduser@hduser-System-Product-Name:~/hadoop$ bin/hadoop-daemon.sh start datanode
starting datanode, logging to /home/hduser/hadoop/bin/../logs/hadoop-hduser-data
node-hduser-System-Product-Name.out
hduser@hduser-System-Product-Name:~/hadoop$ bin/hadoop-daemon.sh start secondary
namenode
starting secondarynamenode, logging to /home/hduser/hadoop/bin/../logs/hadoop-hd
user-secondarynamenode-hduser-System-Product-Name.out
hduser@hduser-System-Product-Name:~/hadoop$ jps
3114 DataNode
2868 NameNode
3039 TaskTracker
3189 SecondaryNameNode
3278 Jps
2960 JobTracker
hduser@hduser-System-Product-Name:~/hadoop$
```

กรณีรันคำสั่ง jps ไม่ได้ ให้แก้ไขโดยรันคำสั่ง

> sudo update-alternatives –install /usr/bin/jps jps /usr/lib/jvm/jdk1.7.0/bin/jps 1

5.3 copy file to HDFS

ก่อนรัน MapReduce ต้องคัดลอกไฟล์ข้อมูลจากหน่วยความจำไปไว้ที่ HDFS โดยรันคำสั่ง

> bin/hadoop dfs -copyFromLocal /home/hduser/tmp/gutenberg

/home/hduser/gutenberg

รูปแบบคำสั่งคัดลอกไฟล์ คือ bin/hadoop dfs -copyFromLocal [ไดเรกทอรี่ที่มีไฟล์ข้อมูล]

[HDFS : ไดเรกทอรี่ปลายทาง]

ดูไฟล์ข้อมูลที่คัดลอกเสร็จแล้วด้วยคำสั่ง

> bin/hadoop dfs -ls /home/hduser/gutenberg

```
hduser@hduser-System-Product-Name:~/hadoop$ bin/hadoop dfs -copyFromLocal /home/hduser/tmp/gutenberg /home/hduser/gutenberg
copyFromLocal: Target /home/hduser/gutenberg/gutenberg is a directory
hduser@hduser-System-Product-Name:~/hadoop$ bin/hadoop dfs -ls /home/hduser
Found 2 items
drwxr-xr-x   - hduser supergroup          0 2013-12-05 01:21 /home/hduser/gutenberg
drwxr-xr-x   - hduser supergroup          0 2013-12-05 00:56 /home/hduser/tmp
```

5.4 รัน MapReduce ในคู่มือนี้ยกตัวอย่างการรัน wordcount เป็นการนับจำนวนคำที่มีในไฟล์ข้อความ

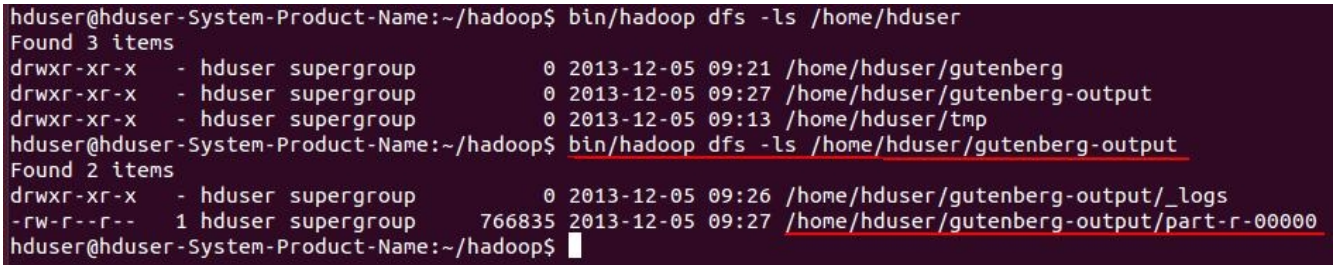
> bin/hadoop jar hadoop-examples-0.20.2.jar wordcount /home/hduser/gutenberg

/home/hduser/gutenberg-output

5.5 ดูผลลัพธ์การรันงาน

> bin/hadoop dfs -ls /home/hduser/gutenberg-output

```
hduser@hduser-System-Product-Name:~/hadoop$ bin/hadoop dfs -ls /home/hduser/gutenberg
Found 3 items
-rw-r--r--   1 hduser supergroup     661808 2013-12-05 09:21 /home/hduser/gutenberg/pg20417.txt
-rw-r--r--   1 hduser supergroup    1540093 2013-12-05 09:21 /home/hduser/gutenberg/pg4300.txt
-rw-r--r--   1 hduser supergroup     794290 2013-12-05 09:21 /home/hduser/gutenberg/pg5000.txt
hduser@hduser-System-Product-Name:~/hadoop$ bin/hadoop jar hadoop-0.20.2-examples.jar wordcount /home/hduser/gut
enberg /home/hduser/gutenberg-output
13/12/05 09:26:40 INFO input.FileInputFormat: Total input paths to process : 3
13/12/05 09:26:40 INFO mapred.JobClient: Running job: job_201312050912_0004
13/12/05 09:26:41 INFO mapred.JobClient:  map 0% reduce 0%
13/12/05 09:26:54 INFO mapred.JobClient:  map 66% reduce 0%
13/12/05 09:27:00 INFO mapred.JobClient:  map 100% reduce 0%
13/12/05 09:27:03 INFO mapred.JobClient:  map 100% reduce 22%
13/12/05 09:27:12 INFO mapred.JobClient:  map 100% reduce 100%
13/12/05 09:27:14 INFO mapred.JobClient: Job complete: job_201312050912_0004
13/12/05 09:27:14 INFO mapred.JobClient: Counters: 17
13/12/05 09:27:14 INFO mapred.JobClient:   Job Counters
13/12/05 09:27:14 INFO mapred.JobClient:     Launched reduce tasks=1
13/12/05 09:27:14 INFO mapred.JobClient:     Launched map tasks=3
13/12/05 09:27:14 INFO mapred.JobClient:     Data-local map tasks=3
13/12/05 09:27:14 INFO mapred.JobClient:   FileSystemCounters
13/12/05 09:27:14 INFO mapred.JobClient:     FILE_BYTES_READ=2012848
13/12/05 09:27:14 INFO mapred.JobClient:     HDFS_BYTES_READ=2996191
13/12/05 09:27:14 INFO mapred.JobClient:     FILE_BYTES_WRITTEN=3285297
13/12/05 09:27:14 INFO mapred.JobClient:     HDFS_BYTES_WRITTEN=766835
13/12/05 09:27:14 INFO mapred.JobClient:   Map-Reduce Framework
13/12/05 09:27:14 INFO mapred.JobClient:     Reduce input groups=71680
13/12/05 09:27:14 INFO mapred.JobClient:     Combine output records=88291
13/12/05 09:27:14 INFO mapred.JobClient:     Map input records=63766
13/12/05 09:27:14 INFO mapred.JobClient:     Reduce shuffle bytes=1005340
13/12/05 09:27:14 INFO mapred.JobClient:     Reduce output records=71680
13/12/05 09:27:14 INFO mapred.JobClient:     Spilled Records=227899
13/12/05 09:27:14 INFO mapred.JobClient:     Map output bytes=5055388
13/12/05 09:27:14 INFO mapred.JobClient:     Combine input records=522098
13/12/05 09:27:14 INFO mapred.JobClient:     Map output records=522098
13/12/05 09:27:14 INFO mapred.JobClient:     Reduce input records=88291
hduser@hduser-System-Product-Name:~/hadoop$
```



5.6 หยุดการทำงาน

> bin/stop-all.sh

การรัน MapReduce บน ubuntu แบบหลายโหนด*

*แก้ไขจากการทำงานแบบ 1 โหนด

1. แก้ไขไฟล์ /etc/hosts

คอมพิวเตอร์ (โหนด) ติดต่อกันผ่านเครือข่ายอินเตอร์เน็ต ทางที่ง่ายที่สุดคือให้ทุกเครื่องวางอยู่ภายในเครือข่ายเดียวกัน โดยตั้งค่า IP ให้แต่ละเครื่อง แล้วแก้ไขไฟล์ /etc/hosts ทำทุกเครื่อง (master, slave)

[หมายเหตุ] ทุกเครื่องที่รัน Hadoop ต้องมีชื่อผู้ใช้ชื่อเดียวกัน (user)

> sudo gedit /etc/hosts

เพิ่มรายละเอียดดังต่อไปนี้

IP ชื่อเครื่อง เช่น

192.168.1.11 gpuubuntu1*

192.168.1.12 gpuubuntu2

...

...

---

*ระบุเท่ากับจำนวนโหนด บรรทัดละ 1 เครื่อง และให้ระบุเหมือนกันทุกเครื่องในไฟล์เดียวกันนี้

2. ติดตั้ง SSH

เครื่องแม่ (master) ต้องสามารถเชื่อมต่อไปยังบัญชีชื่อผู้ใช้ของตัวเอง (user account) และบัญชีชื่อผู้ใช้เดียวกันนี้ในเครื่องลูก (slave) ผ่านการเข้ารหัส SSH เนื่องจากแก้ไขเพิ่มเติมจากการทำแบบ 1 โหนด เพียงแค่รันคำสั่งต่อไปนี้

ทำที่เครื่อง master

> ssh-copy-id -i $HOME/.ssh/id_rsa.pub hduser@IPslave

เช่น ssh-copy-id -i $HOME/.ssh/id_rsa.pub hduser@192.168.1.12

คำสั่งนี้จะแจ้งให้ใส่รหัสผ่านเข้าสู่ระบบสำหรับผู้ใช้ hduser ของ slave จากนั้นคัดลอกคีย์ SSH เพื่อที่เราจะไม่ต้องใส่รหัสผ่านทุกครั้งเมื่อมีการเข้าถึงบัญชีของ Hadoop โดยจะต้องทำเท่ากับจำนวนเครื่อง slave ถ้ามีทั้งหมด 4 เครื่องจะต้องคัดลอกคีย์ทั้งหมด 4 ครั้งที่ IP ต่างกันของแต่ละเครื่อง ทดลองการใช้งานด้วยคำสั่ง

> ssh gpuubuntu2 (ssh ชื่อเครื่อง)

สามารถเข้าถึงบัญชีผู้ใช้ของ Hadoop ที่สร้างไว้ในแต่ละเครื่องได้เลย โดยไม่ต้องใส่รหัสผ่านอีก

> exit (ตัดการเชื่อมต่อ ออกจากบัญชีผู้ใช้)

3. แก้ไขไฟล์ **core-site.xml** ทุกเครื่องและ **mapred-site.xml** ที่ master

เปลี่ยน localhost เป็น ชื่อเครื่อง master ทั้งหมด

นอกจากนี้สามารถกำหนดจำนวน reduce task ได้ในไฟล์ **mapred-site.xml** ที่ master โดยเติมข้อความต่อไปนี้ระหว่าง **<configuration>**

```
<!-- In: conf/mapred-site.xml -->
<property>
  <name>mapred.reduce.tasks</name>
  <value>2</value>
  <description>set num of reduce task.</description>
</property>
```

4. แก้ไขไฟล์ **mapred-site.xml** ที่ slave

เปลี่ยน localhost เป็น ชื่อเครื่อง slave ของแต่ละเครื่องเอง

5. แก้ไขไฟล์ **hdfs-site.xml** ทุกเครื่อง

เปลี่ยน value ให้เท่ากับจำนวนโหนด

นอกจากนี้ยังสามารถกำหนดขนาดบล็อกได้ในไฟล์นี้ (default ขนาดบล็อกคือ 64 M) โดยเติมข้อความต่อไปนี้ระหว่าง **<configuration>** ทำที่ master ถ้าขนาดบล็อกใหญ่ขึ้น จะได้จำนวน map task ลดลง

```
<!-- In: conf/hdfs-site.xml -->
<property>
  <name>dfs.block.size</name>
  <value>134217728</value>
  <description>set dfs block size, 134217728 byte is 128 M</description>
</property>
```

6. กำหนดค่า master และ slave โดยแก้ไขไฟล์ **conf/masters** และ **conf/slaves** ที่ master

6.1 **conf/masters** เติมชื่อเครื่อง master และลบคำว่า localhost

6.2 **conf/slaves** เติมชื่อเครื่องที่ใช้รันงานทั้งหมด 1 บรรทัด 1 เครื่อง และลบคำว่า localhost

7. ลบไฟล์ **conf/masters** และ **conf/slaves** ที่ slave

Running MapReduce

การรันงานแบบหลายโหนดให้รันทุกคำสั่งที่เครื่อง master เท่านั้น

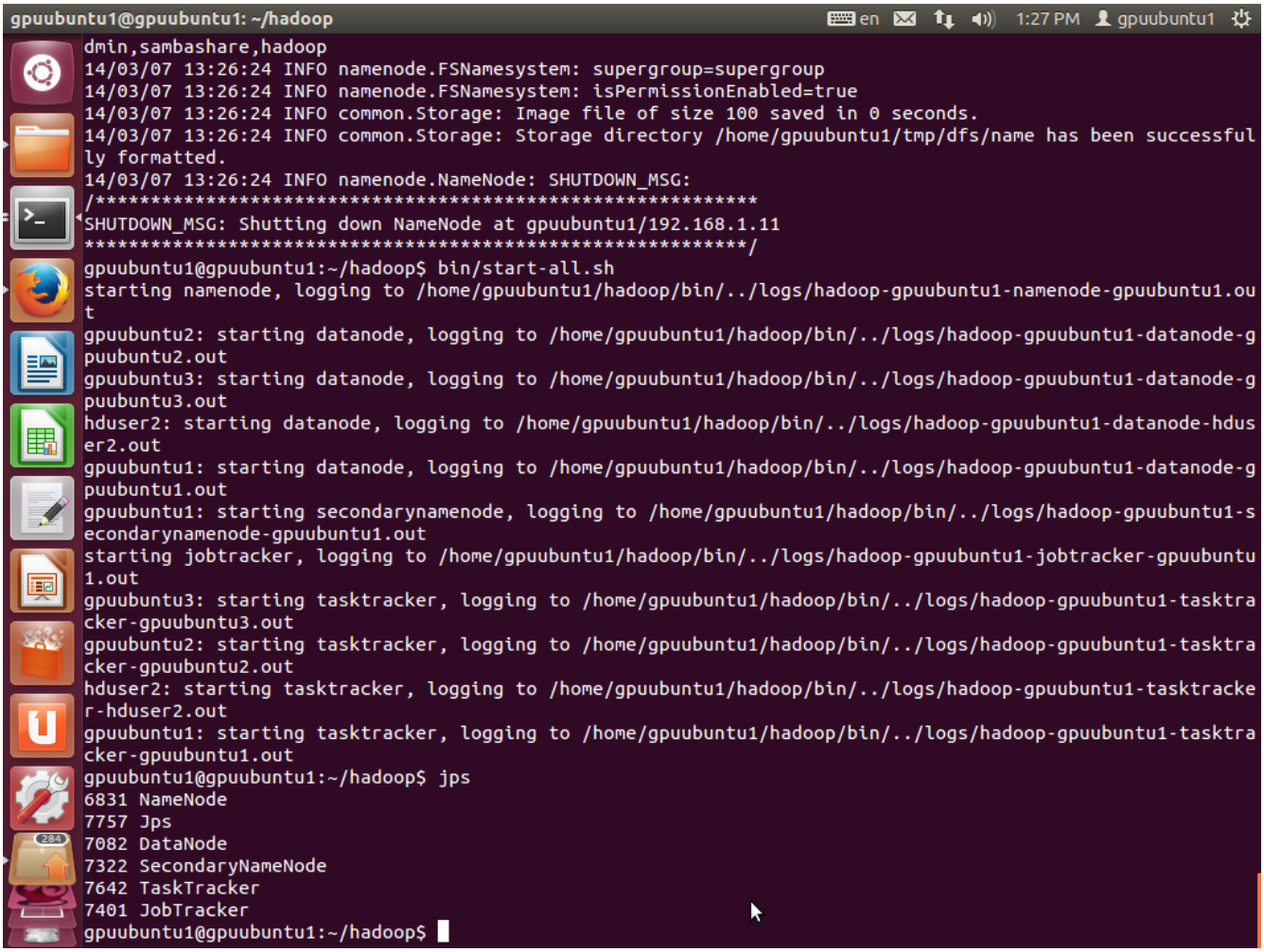
1. ฟอร์แมตเนมโหนด

    > *cd Hadoop*

    > *bin/hadoop namenode -format*

2. สตาร์ตงานทุกประเภทบน **master**

    > *bin/start-all.sh*



งานที่สตาร์ตได้บน **master** คือ Namenode, Datanode, Jobtracker, Tasktracker และ Secondarynamenode

หากพบปัญหาสตาร์ตบางงานไม่ได้ให้รันคำสั่งต่อไปนี้แทน bin/start-all.sh (**รันที่ master**)

> *bin/start-dfs.sh*

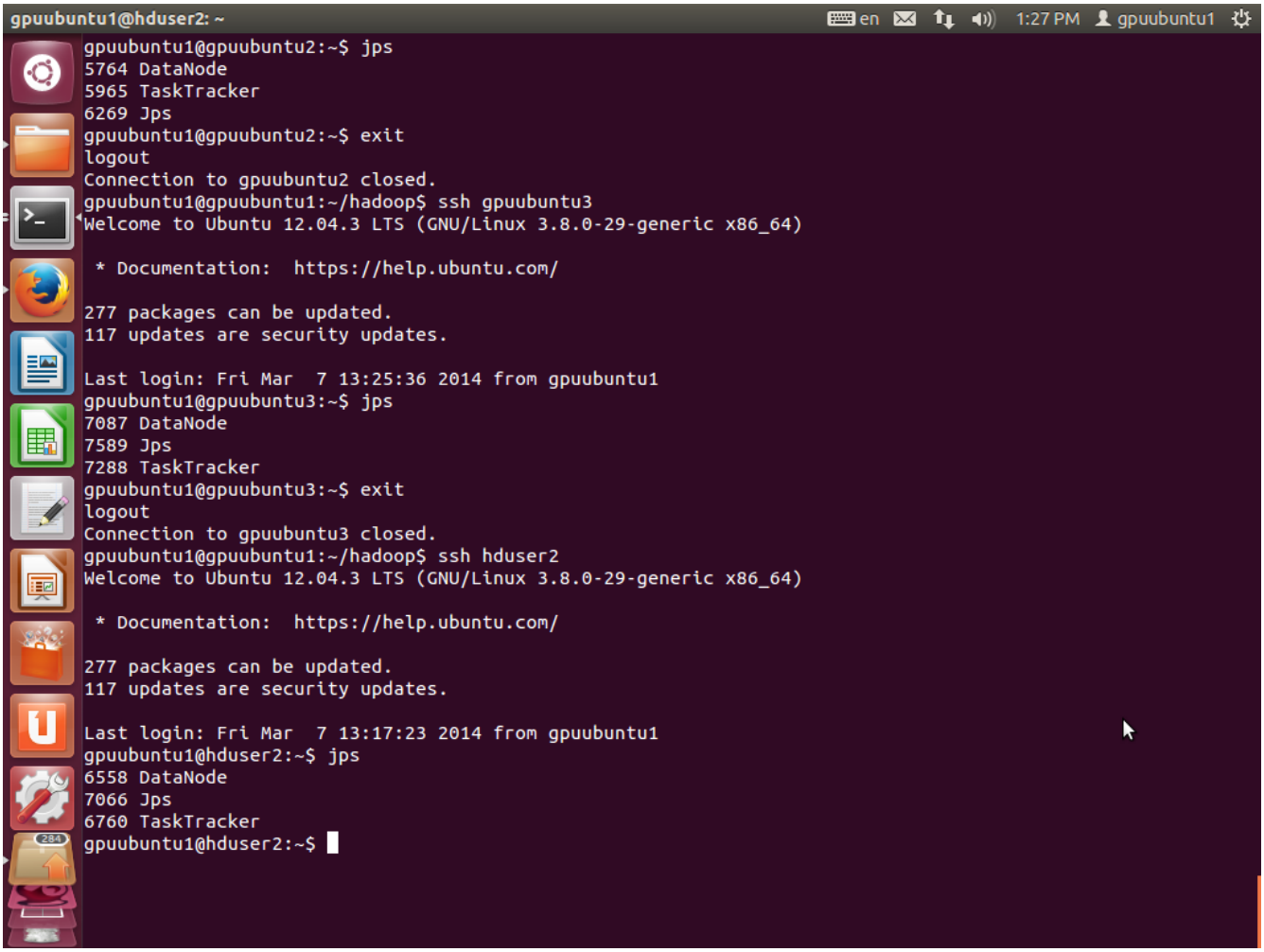
สตาร์ต Namenode, Datanode และ Secondarynamenode บน **master**

สตาร์ต Datanode บน **slave**

> *bin/start-mapred.sh*

สตาร์ต Jobtracker และ Tasktracker บน **master**

สตาร์ต Tasktracker บน **slave**



งานที่สตาร์ตได้บน **slave** คือ Datanode และ Tasktracker

3. copy file to HDFS

> *bin/hadoop dfs -copyFromLocal /home/hduser/Hotel /home/hduser/hdfsHotel*

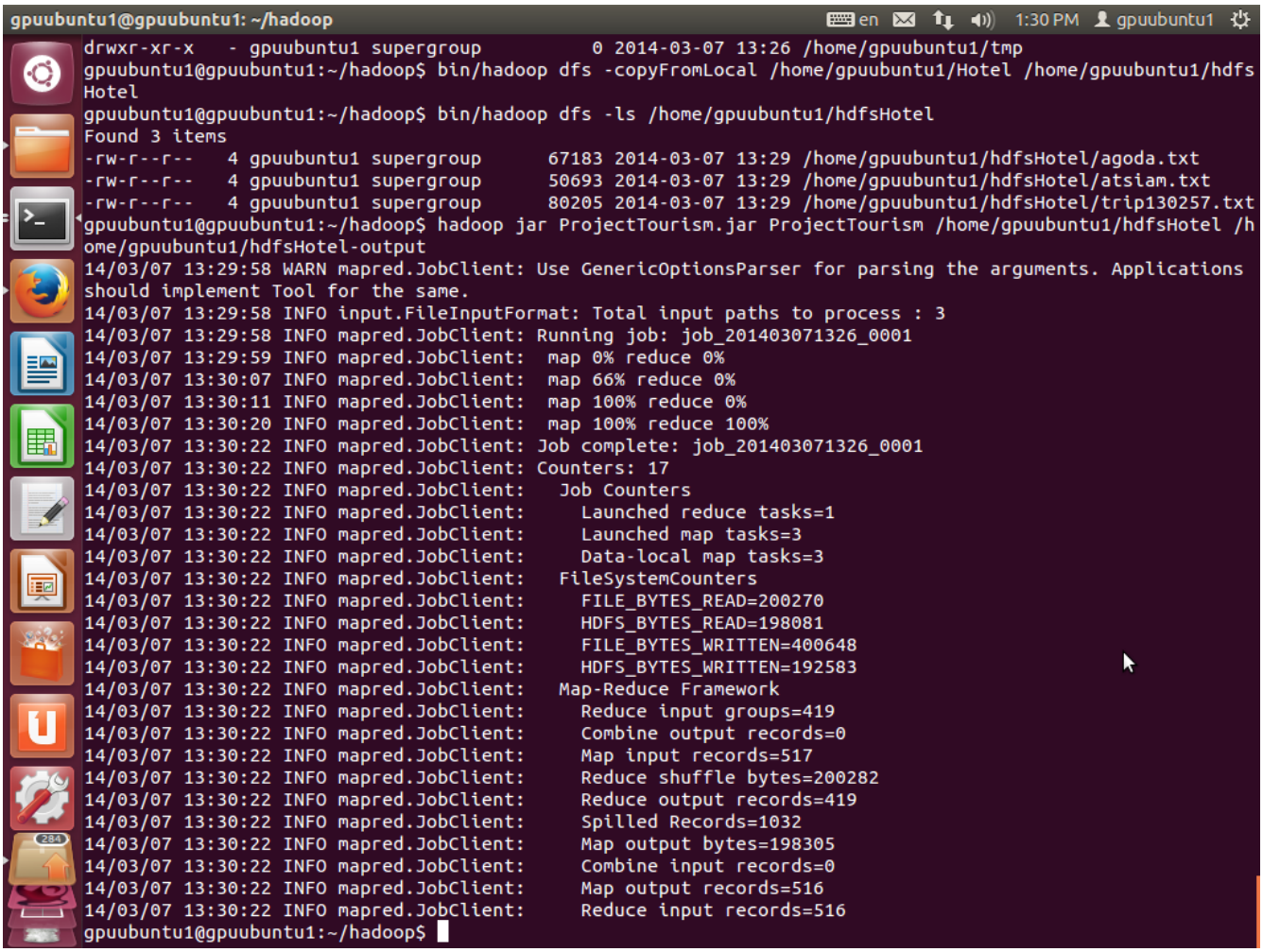
4. running MapReduce

> *hadoop jar ProjectTourism.jar ProjectTourism /home/hduser/hdfsHotel*

/home/hduser/hdfsHotel-output

รูปแบบคำสั่งคือ hadoop jar <jar name.jar> <package.mainclass> <input> <output>

*ยกตัวอย่างการรันงานต่อข้อความแบบ 4 โหนด



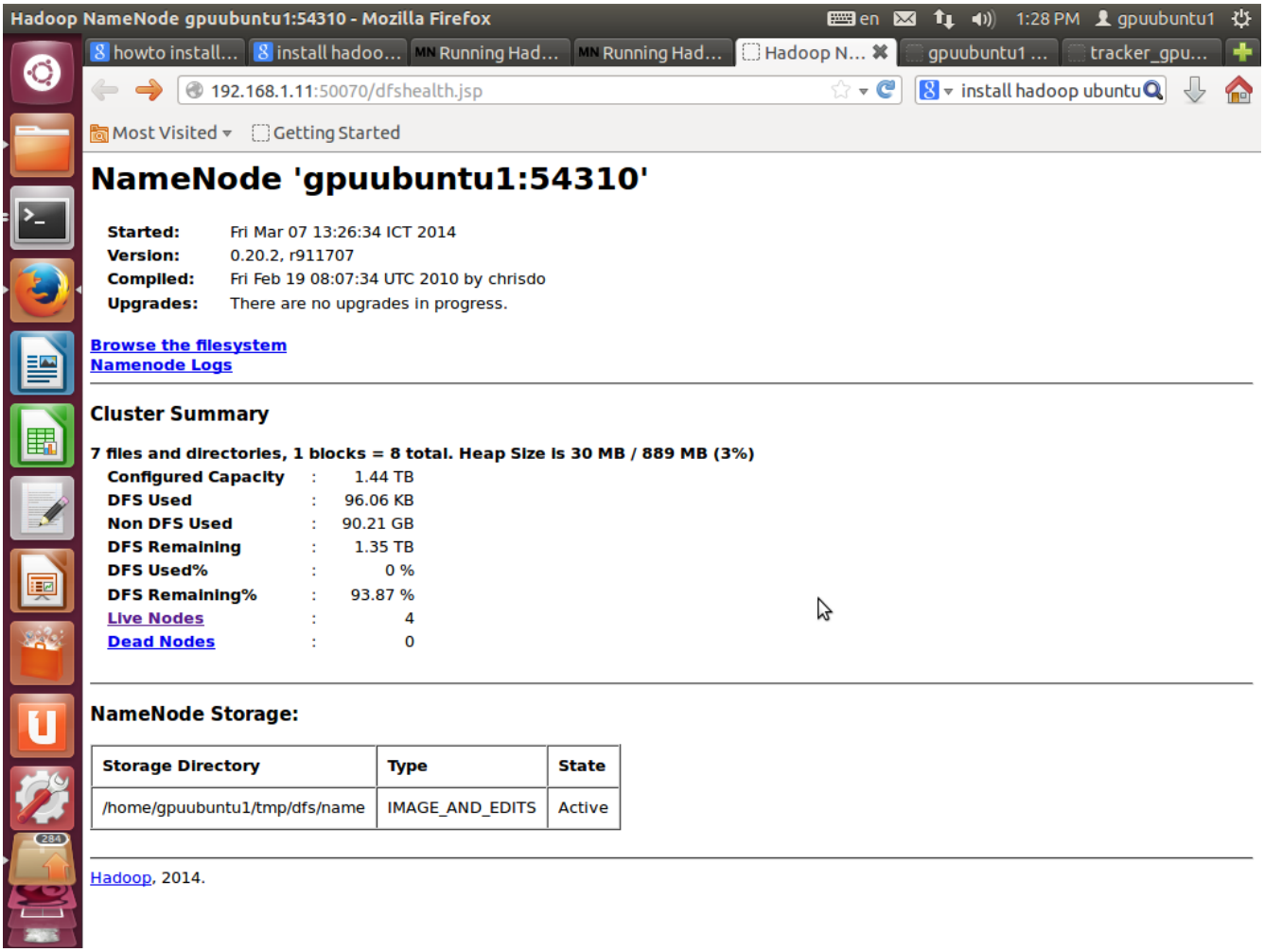
5. หยุดการทำงาน

> *bin/stop-all.sh*

6. ดูผลลัพธ์ผ่านหน้าอินเตอร์เฟซของเครื่อง master

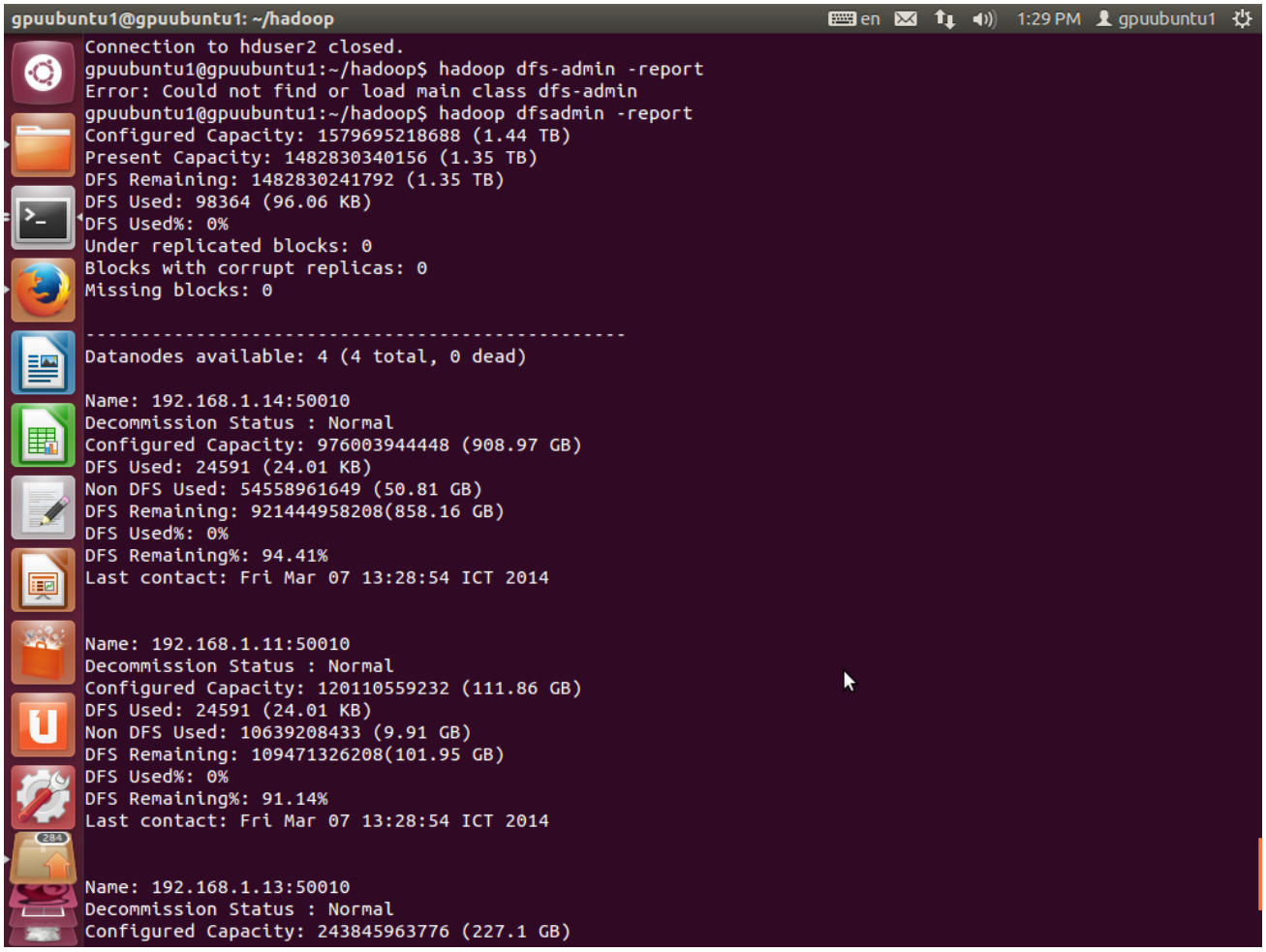
http://IP ของ Master:50070/ = web UI of the NameNode

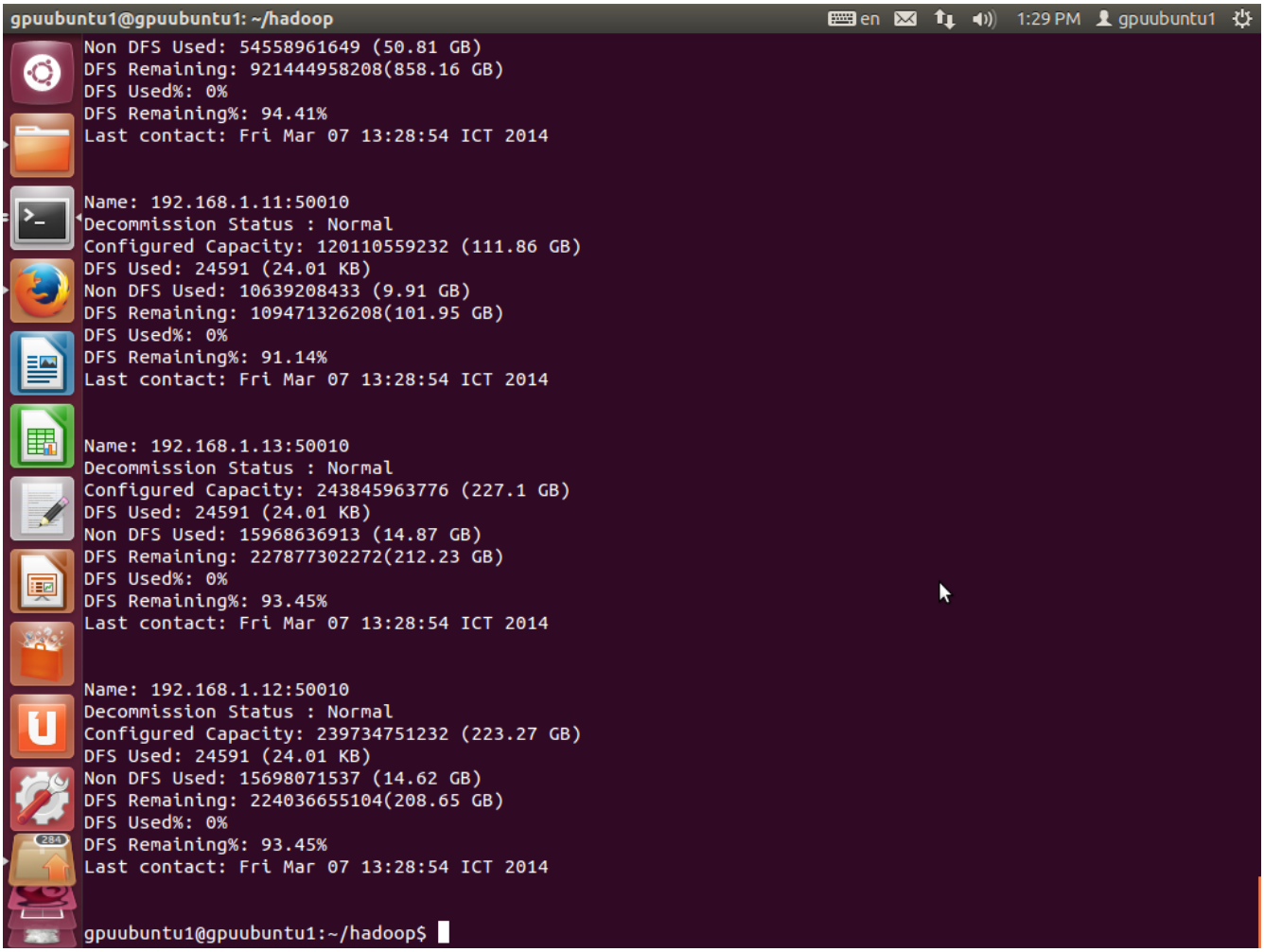
ระบุรายละเอียดหน่วยความจำที่ใช้ จำนวนโหนด ฯลฯ



สามารถรันคำสั่งผ่าน terminal เพื่อดูรายละเอียดของเนมโหนด

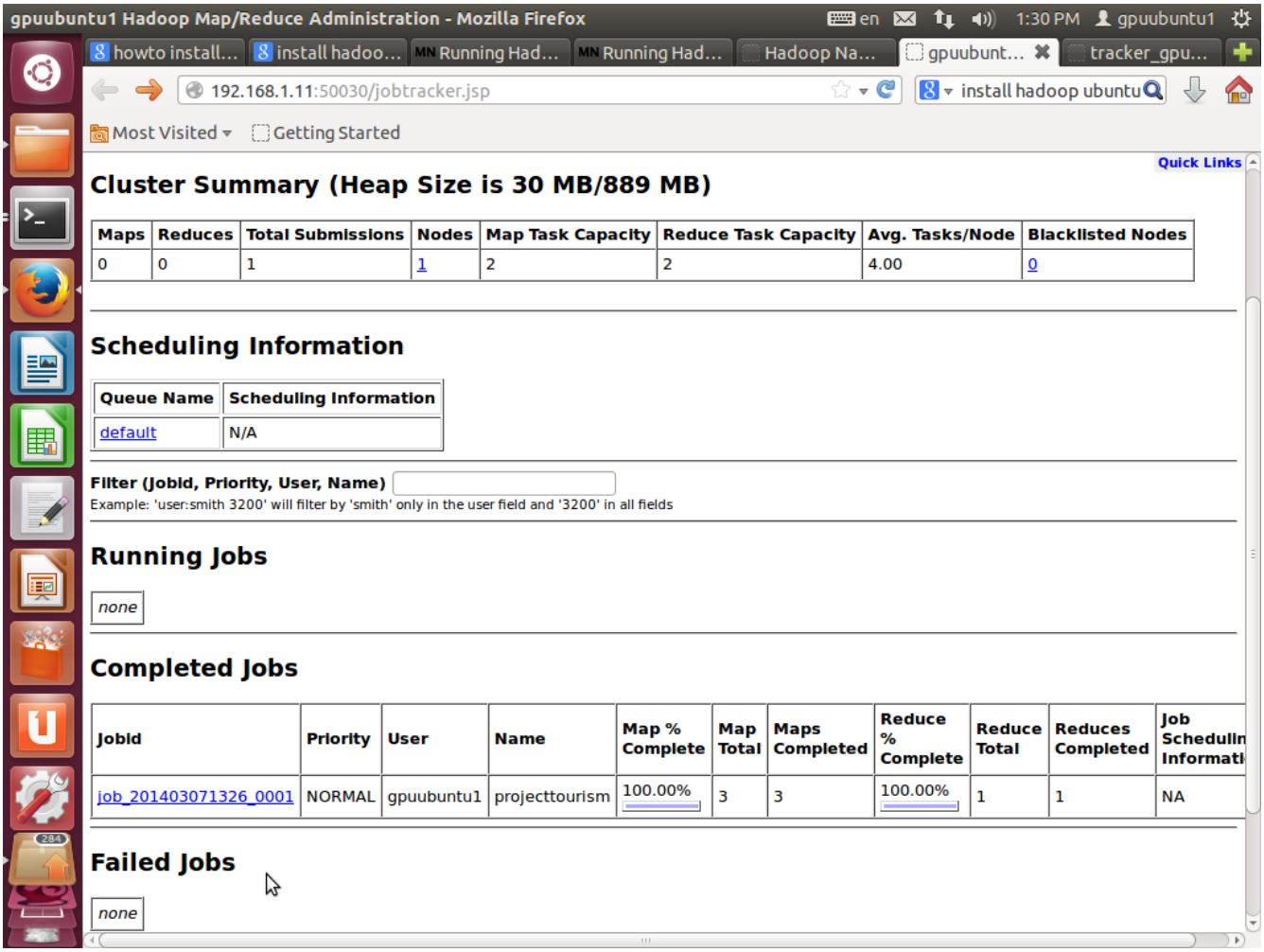
> hadoop dfsadmin -report



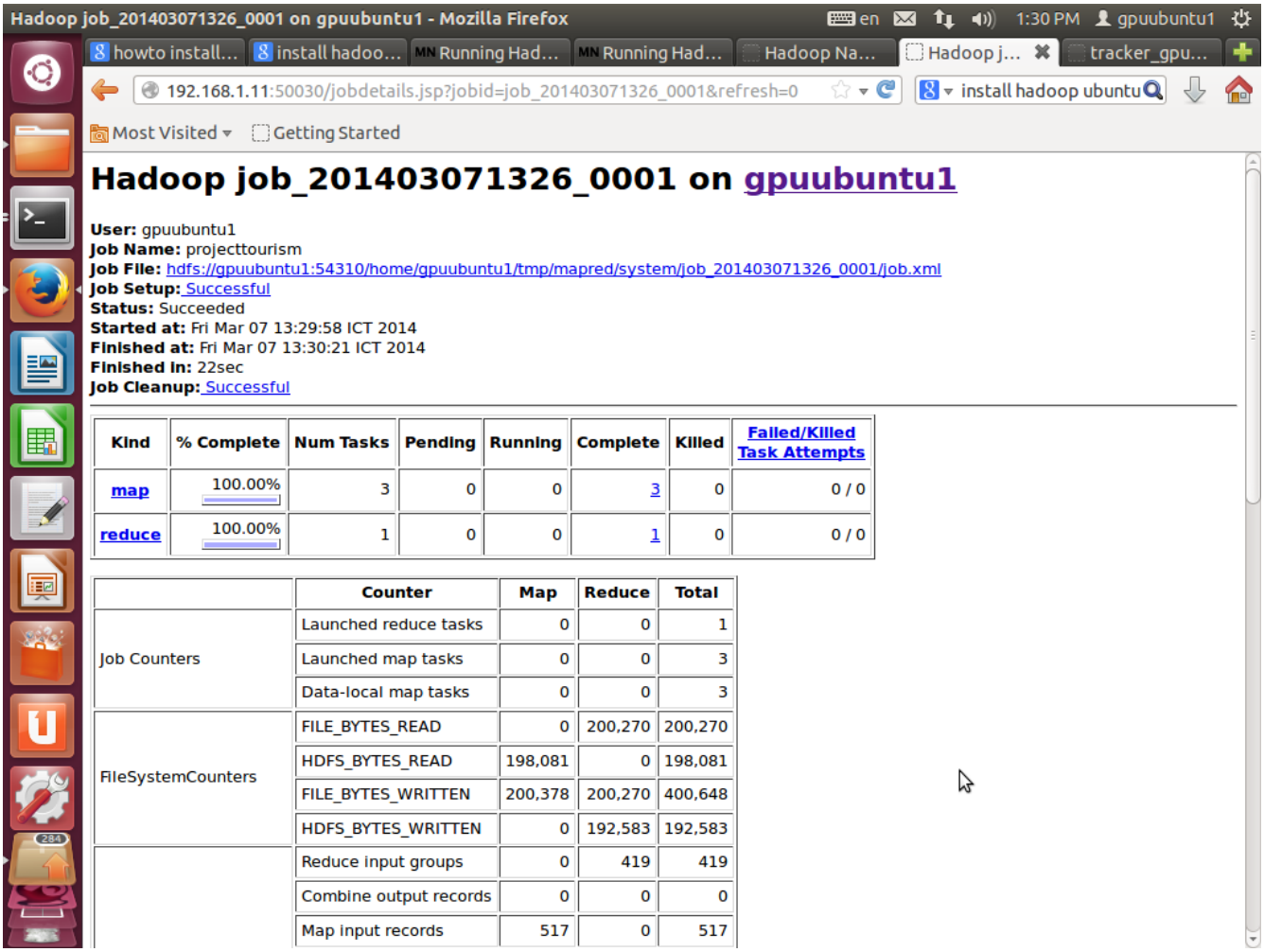

gpuubuntu1@gpuubuntu1: ~/hadoop
en
1:29 PM
gpuubuntu1
Non DFS Used: 54558961649 (50.81 GB)
DFS Remaining: 921444958208(858.16 GB)
DFS Used%: 0%
DFS Remaining%: 94.41%
Last contact: Fri Mar 07 13:28:54 ICT 2014
Name: 192.168.1.11:50010
Decommission Status : Normal
Configured Capacity: 120110559232 (111.86 GB)
DFS Used: 24591 (24.01 KB)
Non DFS Used: 10639208433 (9.91 GB)
DFS Remaining: 109471326208(101.95 GB)
DFS Used%: 0%
DFS Remaining%: 91.14%
Last contact: Fri Mar 07 13:28:54 ICT 2014
Name: 192.168.1.13:50010
Decommission Status : Normal
Configured Capacity: 243845963776 (227.1 GB)
DFS Used: 24591 (24.01 KB)
Non DFS Used: 15968636913 (14.87 GB)
DFS Remaining: 227877302272(212.23 GB)
DFS Used%: 0%
DFS Remaining%: 93.45%
Last contact: Fri Mar 07 13:28:54 ICT 2014
Name: 192.168.1.12:50010
Decommission Status : Normal
Configured Capacity: 239734751232 (223.27 GB)
DFS Used: 24591 (24.01 KB)
Non DFS Used: 15698071537 (14.62 GB)
DFS Remaining: 224036655104(208.65 GB)
DFS Used%: 0%
DFS Remaining%: 93.45%
Last contact: Fri Mar 07 13:28:54 ICT 2014
gpuubuntu1@gpuubuntu1:~/hadoop$

http://IP ของ Master:50030/ = web UI of the JobTracker

ระบุชื่อเครื่องที่รัน ชื่องาน เปอร์เซ็นการทำงาน ฯลฯ



เมื่อคลิกเข้าไปที่ JobId จะระบุรายละเอียดงานเพิ่มขึ้นอีก



Hadoop job_201403071326_0001 on gpuubuntu1

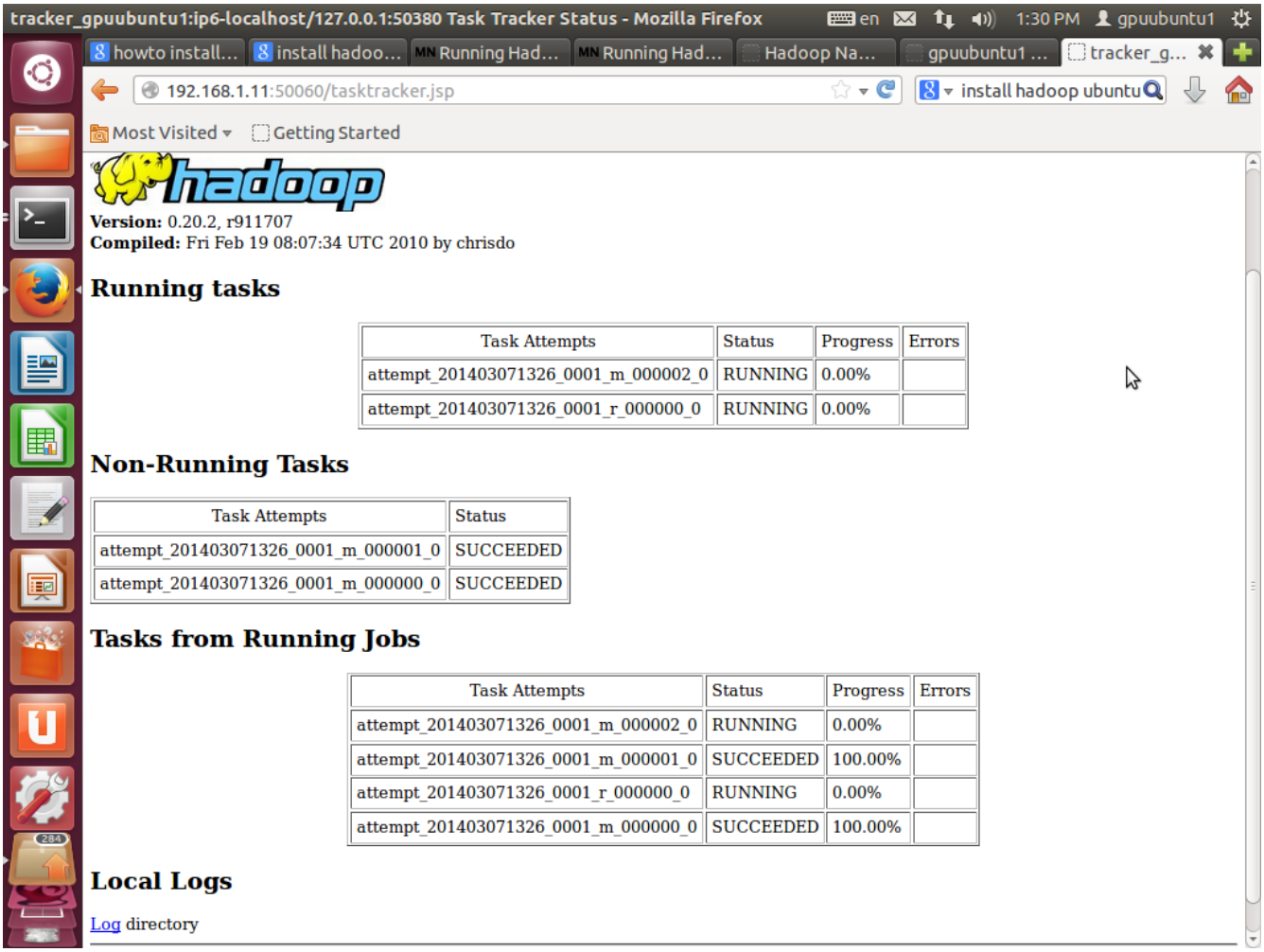
User: gpuubuntu1
Job Name: projecttourism
Job File: hdfs://gpuubuntu1:54310/home/gpuubuntu1/tmp/mapred/system/job_201403071326_0001/job.xml
Job Setup: Successful
Status: Succeeded
Started at: Fri Mar 07 13:29:58 ICT 2014
Finished at: Fri Mar 07 13:30:21 ICT 2014
Finished in: 22sec
Job Cleanup: Successful

Kind	% Complete	Num Tasks	Pending	Running	Complete	Killed	Failed/Killed Task Attempts
map	100.00%	3	0	0	3	0	0 / 0
reduce	100.00%	1	0	0	1	0	0 / 0

	Counter	Map	Reduce	Total
Job Counters	Launched reduce tasks	0	0	1
	Launched map tasks	0	0	3
	Data-local map tasks	0	0	3
FileSystemCounters	FILE_BYTES_READ	0	200,270	200,270
	HDFS_BYTES_READ	198,081	0	198,081
	FILE_BYTES_WRITTEN	200,378	200,270	400,648
	HDFS_BYTES_WRITTEN	0	192,583	192,583
	Reduce input groups	0	419	419
	Combine output records	0	0	0
	Map input records	517	0	517

http://IP ของ Master:50060/ = web UI of the TaskTracker

ระบุงานที่กำลังทำ



ผลการทดลองแสดงเวลาการทำงาน

เวลาการทำงานวัดจากการรัน map บวกกับการรัน reduce

1. รัน wordcount 1 โหนดของไฟล์ขนาด 4 GB จำนวน 1 ไฟล์

เครื่องที่ใช้รัน

- core = 4
- memory = 7.7 GiB
- disk = 976.0 GB

1 Node	Time				
Block Size (M)	**64**	**128**	**256**	**512**	**1024**
Task	**Map = 61**	**Map = 31**	**Map = 16**	**Map = 8**	**Map = 4**
Reduce = 1	5 mins, 21 sec	5 mins, 5 sec	4 mins, 45 sec	4 mins, 26 sec	4 mins, 23 sec
Reduce = 2	5 mins, 19 sec	5 mins, 4 sec	4 mins, 33 sec	4 mins, 15 sec	4 mins, 14 sec

2. รัน wordcount 4 โหนดของไฟล์ขนาด 4 GB จำนวน 1 ไฟล์

เครื่องที่เป็น master

- core = 4
- memory = 7.7 GiB
- disk = 976.0 GB

4 Node	Time				
Block Size (M)	**64**	**128**	**256**	**512**	**1024**
Task	**Map = 61**	**Map = 31**	**Map = 16**	**Map = 8**	**Map = 4**
Reduce = 1	5 mins, 22 sec	4 mins, 45 sec	4 mins, 42 sec	4 mins, 16 sec	4 mins, 15 sec
Reduce = 2	5 mins, 10 sec	4 mins, 38 sec	4 mins, 27 sec	4 mins, 12 sec	4 mins, 14 sec

3. รัน dbpedia (concat string) 1 โหนดของไฟล์ขนาด 4 GB จำนวน 1 ไฟล์

เครื่องที่ใช้รัน

- core = 4
- memory = 7.7 GiB
- disk = 976.0 GB

1 Node	Time				
Block Size (M)	**64**	**128**	**256**	**512**	**1024**
Task	**Map = 61**	**Map = 31**	**Map = 16**	**Map = 8**	**Map = 4**
Reduce = 1	7 mins, 35 sec	7 mins, 31 sec	6 mins, 40 sec	6 mins, 2 sec	6 mins, 48 sec
Reduce = 2	7 mins, 10 sec	7 mins, 23 sec	6 mins, 29 sec	5 mins, 40 sec	6 mins, 12 sec

4. รัน dbpedia (concat string) 4 โหนดของไฟล์ขนาด 4 GB จำนวน 1 ไฟล์

เครื่องที่ใช้รัน

- core = 4
- memory = 7.7 GiB
- disk = 976.0 GB

4 Node	Time				
Block Size (M)	**64**	**128**	**256**	**512**	**1024**
Task	**Map = 61**	**Map = 31**	**Map = 16**	**Map = 8**	**Map = 4**
Reduce = 1	7 mins, 9 sec	7 mins, 10 sec	6 mins, 49 sec	6 mins, 35 sec	6 mins, 31 sec
Reduce = 2	6 mins, 50 sec	7 mins, 30 sec	6 mins, 12 sec	5 mins, 22 sec	6 mins, 2 sec

ปัจจัยที่มีผลต่อเวลาการทำงาน

1. จำนวน data disk

    จำนวนเพิ่มขึ้น เวลาการทำงานลดลง

2. การบีบอัดเอาต์พุตของ map

    เมื่อบีบอัดเอาต์พุต เวลาการทำงานลดลง

3. ขนาดบล็อกข้อมูล

    ขนาดบล็อกมาก เวลาการทำงานลดลง

4. จำนวน map

    จำนวนงานน้อย เวลาการทำงานลดลง

5.จำนวน reduce

    จำนวนงานมาก เวลาการทำงานลดลง

การปรับพารามิเตอร์ใน Hadoop

hdfs-site.xml

พารามิเตอร์	คำอธิบาย
dfs.block.size	ขนาดบล็อกข้อมูล (byte)
dfs.name.dir	ไดเรกทอรี่เก็บข้อมูล namenode
dfs.data.dir	ไดเรกทอรี่เก็บข้อมูล datanode
dfs.datanode.handler.count	จำนวน server threads ของ datanode
dfs.namenode.handler.count	จำนวน server threads ของ namenode
dfs.replication	แบบจำลองจำนวนบล็อก
dfs.replication.max	แบบจำลองจำนวนบล็อกสูงสุด
dfs.namenode.replication.min	แบบจำลองจำนวนบล็อกต่ำสุด
dfs.client.block.write.retries	จำนวนการลองเขียนบล็อกที่ datanode
dfs.namenode.max.objects	จำนวนสูงสุดของไฟล์ ไดเรกทอรี่ และบล็อก
dfs.namenode.replication.interval	ระยะเวลาที่ namenode คำนวณงานให้ datanode (second)

mapred-site.xml

พารามิเตอร์	คำอธิบาย
mapred.compress.map.output	บีบอัดเอาท์พุตของ map ก่อนส่งผ่านเครือข่าย
mapred.map.tasks.speculative.execution	กำหนดงาน map รันแบบขนาน
mapred.reduce.tasks.speculative.execution	กำหนดงาน reduce รันแบบขนาน
mapred.tasktracker.map.tasks.maximum	จำนวนงานสูงสุดของ map ที่รันโดย TaskTracker
mapred.tasktracker.reduce.tasks.maximum	จำนวนงานสูงสุดของ reduce ที่รันโดย TaskTracker
io.sort.mb	ขนาดหน่วยความจำบัฟเฟอร์ของ map ที่ใช้เรียงเอาต์พุต (megabytes)
mapred.job.reuse.jvm.num.tasks	จำนวนงานที่รันผ่าน JVM
mapred.reduce.parallel.copies	จำนวนเธรดที่ใช้คัดลอกผลของ map ไปสู่การ reduce
mapred.min.split.size	ขนาดต่ำสุดในการแบ่งอินพุตของ map
mapred.max.split.size	ขนาดสูงสุดในการแบ่งอินพุตของ map
io.sort.spill.percent	Spill เนื้อหาไปสู่ดิสก์
mapred.local.dir	ไดเรกทอรี่เก็บข้อมูลการทำงาน
mapred.job.shuffle.input.buffer.percent	เปอร์เซ็นต์หน่วยความจำที่ใช้เรียงลำดับเอาต์พุตของ map ระหว่างการ shuffle
mapred.job.shuffle.merge.percent	เปอร์เซ็นต์หน่วยความจำที่ใช้จัดเก็บเอาต์พุตของ map
mapred.task.timeout	ระยะเวลาในการอ่าน เขียน และแก้ไขข้อมูล (millisecond)
io.sort.record.percent	เปอร์เซ็นต์ของ io.sort.mb
mapred.job.tracker.handler.count	จำนวน server threads ของ JobTracker
mapred.max.tracker.failures	จำนวนงานที่ล้มเหลวบน TaskTracker
mapred.map.tasks	จำนวน map
mapred.reduce.tasks	จำนวน reduce
mapred.job.reduce.input.buffer.percent	เปอร์เซ็นต์หน่วยความจำที่ใช้เก็บเอาต์พุตของ map

mapred..child.java.opts	Java opts ของ TaskTracker
mapred.job.tracker	host และ port ที่ติดตามงานของ MapReduce

ภาคผนวก ข คำค้นในการทดสอบประสิทธิภาพด้านเวลา

ตารางที่ คำค้นที่ใช้ในการทดลองเพื่อวัดเวลาในการสืบค้นขั้นสูง

ลำดับที่	กำหนดเงื่อนไขการค้น	จำนวนผลลัพธ์	เวลาในการคิวรี่ (s)
1	Cha-Am, Day Spa	20	0.063963136
2	Cha-Am, Day Spa, Facial Mask, Detox Massage	0	0.068681728
3	Hua Hin, Day Spa, Clairvoyant	1	0.061341696
4	Hua Hin, Hotel & Resort Spa, Clairvoyant, Cleansing	0	0.055574528
5	Cha-Am, Hua Hin, Clairvoyant, Aroma Massage	26	0.061341696
6	Cha-Am, Day Spa, Clairvoyant, Aroma Massage, Body Massage	9	0.066060288
7	Cha-Am, Hua Hin, Clairvoyant, Aroma Massage, Collagen Massage, Free Service	2	0.06815744
8	Cha-Am, Hua Hin, Gold, Face Massage	1	0.06815744
9	Cha-Am, Hua Hin, Aroma Massage, Body Massage, Collagen Massage,<500	18	0.07340032
10	Cha-Am, Hua Hin, Facial Mask, Herbal Steam, Manicure, Skin Scrub, Spa Hand & Foot	34	0.065011712
11	Cha-Am, Hua Hin, Hotel & Resort Spa, Day Spa, Cleansing, Facial Mask, Foot Massage, WIFI	0	0.061865984
12	Cha-Am, Hua Hin, Day Spa, Aroma Massage, Hot Oil Massage, Neck Shoulder Back Massage, <5000	25	0.072351744
13	Hua Hin, Day Spa, Head Massage	2	0.065011712
14	Hua Hin, Head Massage, Hot Oil Massage, Neck Shoulder Back Massage, Nature Massage	19	0.06291456
15	Hua Hin, Hotel & Resort Spa, Clairvoyant, Cleansing, Slimming Massage, Free Service, WIFI, Single Room	0	0.069206016
16	Hua Hin, Free Service, WIFI, Gym, Sauna	16	0.054525952

ลำดับที่	กำหนดเงื่อนไขการค้น	จำนวนผลลัพธ์	เวลาในการคิวรี่ (s)
17	Hua Hin, Skin Scrub, Waxing, Body Massage, Detox Massage	20	0.061865984
18	Hua Hin, Slimming Massage, WIFI	0	0.061865984
19	Hua Hin, Spa Hand & face, Treatment Face, Waxing, Aroma Massage	22	0.053477376
20	Cha-Am, Hotel & Resort, Herbal Steam, Treatment Face, Face Massage, Head Massage, General Room, Yoga Room, <2000	0	0.067108864
21	Hua Hin, Destination, Treatment Face, Face Massage, WIFI	0	0.066060288
22	Hua Hin, Day Spa, Sliver, Collagen Massage	0	0.056623104
23	Hua Hin, Aroma Massage, Coconut Oil Massage, Collagen Massage	19	0.05767168
24	Cha-Am, Hua Hin, Facial Mask, Treatment Face, Hot Oil Massage,<500	5	0.056623104
25	Cha-Am, Hua Hin, Head Massage, <500	0	0.054525952
26	Cha-Am, Body Massage, Foot Massage	18	0.049283072
27	Cha-Am, Sliver	0	0.065011712
28	Cha-Am, Hua Hin, Day Spa, Detox Massage, Hot Oil Massage, Hot Stone Massage	0	0.07340032
29	Hua Hin, Destination, Herbal Steam, Head Massage	0	0.050331648
30	Hua Hin, Cleansing, Facial Mask, Herbal Steam, Manicure, Waxing	16	0.058720256
31	Hua Hin, Aroma Massage, Detox Massage, Face Massage, Nature Massage, Slimming Massage	20	0.06291456
32	Hua Hin	156	0.04194304
33	Hua Hin, Destination	3	0.04194304
34	Hua Hin, Hotel & Resort Spa	31	0.048234496
35	Hua Hin, Hotel& Resort Spa, Gold, Sliver	6	0.054525952

ลำดับที่	กำหนดเงื่อนไขการค้น	จำนวนผลลัพธ์	เวลาในการคิวรี่ (s)
36	Cha-Am, Hua Hin, Facial Mask, Free Service, <500	0	0.06291456
37	Hua Hin, Manicure, Skin Scrub, Collagen Massage, Detox Massage	22	0.050331648
38	Hua Hin, Aroma Massage, Body Massage, Coconut Oil Massage, Collagen Massage, Detox Massage, Face Massage, Foot Massage, Head Massage, Hot Oil Massage	29	0.058720256
39	Hua Hin, Body Massage, Detox Massage, Single Room, Yoga Room	2	0.060817408
40	Hua Hin, Body Massage, Face Massage, Free Service, WIFI, Gym, Sauna	2	0.06291456
41	Cha-Am, Skin Scrub, Aroma Massage	13	0.054525952
42	Hua Hin, Cleansing, Facial Mask, Waxing, Detox Massage, Head Massage, Hot oil, Massage, Nature Massage	14	0.069206016
43	Cha-Am, Free Service, Gym, Locker, Yoga Room	1	0.054525952
44	Hua Hin, Head Massage, Skin Scrub, < 1000	12	0.056623104
45	Cha-Am, Day Spa, Treatment Face	0	0.054525952
46	Hua Hin, Body Massage, <1000	0	0.058720256
47	Hua Hin, Head Massage, Foot Massage	27	0.048234496
48	Hua Hin, Herbal Steam, Manicure, Skin Scrub, Aroma Massage, Body Massage, Coconut Oil Massage, Collagen Massage, Detox Massage, Face Massage	26	0.067108864
49	Hua Hin, Free Service, WIFI, Gym, Sauna, <2000	3	0.06291456
50	Cha-Am, Hotel & Resort, Silver	0	0.046137344
51	Hua Hin, Hotel & Resort, Gold, Nature Massage	0	0.171835392

ลำดับที่	กำหนดเงื่อนไขการค้น	จำนวนผลลัพธ์	เวลาในการคิวรี่ (s)
52	Cha-Am, Hua Hin, Gold, Hot Oil Massage, Hot Stone Massage, Single Room	1	0.194248704
53	Hua Hin, Detox Massage, Face Massage, General Room	2	0.074448896
54	Hua Hin, Hotel & Resort Spa, Sauna	11	0.064487424
55	Cha-Am, Hotel& Resort Spa, Detox Massage, Yoga Room	0	0.077856768
56	Cha-Am, Hua Hin, Day Spa, Manicure, <500	10	0.135266304
57	Cha-Am, Hua Hin, Day Spa, Spa Hand & Foot, Aroma Massage	26	0.062390272
58	Hua Hin, Destination Spa, Single Room	0	0.07077888
59	Hua Hin, Detox Massage, Free Service	0	0.062390272
60	Hua Hin, Facial Mask, Detox Massage, Nature Massage	7	0.058195968
61	Hua Hin, Day Spa, Neck Shoulder Back Massage, Nature Massage	16	0.06291456
62	Hua Hin, Day Spa, Manicure, Spa Hand & Foot	12	0.058195968
63	Hua Hin, Day Spa, Treatment Face, Collagen Massage	7	0.060817408
64	Hua Hin, Day Spa, Hot Oil Massage, Neck Shoulder Back Massage, <2000	13	0.062390272
65	Cha-Am, Day Spa, Manicure, Foot Massage	16	0.052953088
66	Cha-Am, Hua Hin, Cleansing, Sauna, Yoga Room	0	0.058195968
67	Hua Hin, Hotel & resort Spa, Gold	4	0.055574528
68	Hua Hin, Hotel & resort, Hot Stone Massage	1	0.054001664
69	Hua Hin, Detox Massage, Nature Massage, Single Room	1	0.060817408
70	Hua Hin, Face Massage, Foot Massage, Head Massage, Hot Oil Massage, Neck Shoulder Back Massage, <1000	22	0.061341696

ลำดับที่	กำหนดเงื่อนไขการค้น	จำนวนผลลัพธ์	เวลาในการคิวรี่ (s)
71	Hua Hin, Aroma Massage	19	0.04980736
72	Hua Hin, Collagen Massage, Hot oil massage	2	0.04980736
73	Hua Hin, Foot Massage	26	0.04980736
74	Cha- Am, Hua Hin, Gym	6	0.050331648
75	Cha-Am, Body Massage, WIFI	0	0.057147392
76	Hua Hin, Skin Scrub, Body Massage	20	0.044040192
77	Cha-Am, Hua Hin, Face Massage, Foot Massage, <1000	31	0.054525952
78	Cha-Am, Hua Hin, Slimming Massage, General room	0	0.055574528
79	Hua Hin, Day Spa, Body Massage, Neck Shoulder Back Massage	15	0.053477376
80	Hua Hin, Skin Scrub, Spa Hand & Foot, Treatment Face, Waxing	19	0.0524288
81	Hua Hin, Sauna, WIFI	12	0.044040193
82	Hua Hin, Aroma Massage, Face Massage, Foot Massage, Head Massage	29	0.051380224
83	Cha-Am, Hua Hin, Cleansing, Facial Mask, Face Massage	7	0.055574528
84	Cha-Am, Body Massage, Coconut Massage, <1000	0	0.054525952
85	Hua Hin, Aroma Massage, Body Massage, Neck Shoulder Back Massage, Nature massage	23	0.050331648
86	Hua Hin, Waxing	7	0.061865984
87	Cha-Am, Hua Hin	179	0.042991616
88	Hua Hin, Herbal Steam, Coconut Oil Massage	2	0.050331648
89	Hua Hin, Treatment Face, Single Room	2	0.051380224
90	Cha-Am, Hua Hin, Manicure, General Room	13	0.0524288
91	Hua Hin, Facial Mask, General Room	1	0.05767168

ลำดับที่	กำหนดเงื่อนไขการค้น	จำนวนผลลัพธ์	เวลาในการคิวรี่ (s)
92	Cha-Am, Hua Hin, Skin Scrub, Treatment Face, General room	21	0.051380224
93	Hua Hin, Day Spa, Hotel & Resort Spa, Detox Massage	2	0.122683392
94	Hua Hin, Collagen Massage, Free Service	0	0.050331648
95	Cha- Am, Hua Hin, Foot Massage, Slimming Massage	45	0.050331648
96	Hua Hin, Hot Oil Massage, WIFI	0	0.050331648
97	Hua Hin, Hotel & Resort, Body Massage	1	0.045088768
98	Hua Hin, Coconut Oil Massage, Collagen Massage, Detox Massage	3	0.045088768
99	Hua Hin, Face Massage, <1000	0	0.056623104
100	Cha-Am, Hua Hin, Body Massage, Collagen Massage	5	0.053477376
	เวลาเฉลี่ย	**13**	**0.061618**