

ระบบค้นหาสินค้าลดราคาด้วยวิธีการเข้าถึงข้อมูลหน้าเว็บไซต์

Find items for sale with web scrapping

ประหยัด เลวัน และ ชงชัย เทียงธรรม

สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยรามคำแหง

Email: prayatl@gmail.com, mandarinkb@gmail.com

บทคัดย่อ

แอปพลิเคชันค้นหาสินค้าลดราคา จัดทำเพื่อเป็นเครื่องมือช่วยค้นหาสินค้าลดราคาจากเว็บไซต์ต่างๆ ซึ่งใช้วิธีการเข้าถึงข้อมูลหน้าเว็บไซต์ (web scrapping) โดยได้แบ่งการทำงานออกเป็น 2 ส่วน คือ ส่วนที่หนึ่ง ระบบเข้าถึงข้อมูลหน้าเว็บไซต์ต่างๆ ระบบนี้จะทำการดึงข้อมูลหน้าเว็บไซต์ต่างๆ ที่ต้องการ จากนั้นทำการจัดเก็บข้อมูลลงใน NoSQL Database ที่มีชื่อว่า Elasticsearch ส่วนที่สอง เป็นส่วนแสดงผล โดยแสดงผ่านแอปพลิเคชันค้นหาสินค้าลดราคา บนระบบปฏิบัติการแอนดรอยด์ การพัฒนาระบบค้นหาสินค้าลดราคาด้วยวิธีการเข้าถึงข้อมูลเว็บไซต์ มีเครื่องมือที่ช่วยพัฒนาได้แก่ โปรแกรม Virtual Studio Code โปรแกรม Android Studio โปรแกรม Postman โปรแกรมภาษาจาวา (Java) โปรแกรม Redis โปรแกรม Elasticsearch และโปรแกรม Docker ผลการทดสอบการดำเนินงานของระบบพบว่า ค่าเฉลี่ยผลความครบถ้วนของข้อมูลที่ทำ การสแครปป์มาจากเว็บไซต์ร้านค้าออนไลน์ขนาดใหญ่ ร้อยละ 99.95 และเมื่อเพิ่มจำนวนบอทจาก 3 ไปเป็น 13 บอทเวลาประมวลผลลดลงถึงประมาณ 10 เท่า (170.24/17.21) ในส่วนแอปพลิเคชันค้นหาสินค้าลดราคา ผลการทดสอบเมนูค้นหาพบว่าการค้นพบสินค้าตามที่ต้องการคิดเป็นค่าเฉลี่ย 81.30 % ผลสรุปด้านประสิทธิภาพของแอปพลิเคชันประกอบด้วยเนื้อหา 10 หัวข้อ โดยได้ให้ผู้ใช้และผู้พัฒนาระบบทดลองใช้งานผลปรากฏว่า ผู้ใช้มีความพึงพอใจอยู่ในระดับ “มาก” ($\bar{X} = 4.20$) และผู้พัฒนาระบบ

มีความพึงพอใจในมุมมองการออกแบบฟังก์ชันการใช้งานอยู่ในระดับ “มาก” ($\bar{X} = 4.20$) และผลการทดสอบเวิร์กเวอร์สามารถรองรับการใช้งาน 270 ผู้ใช้งานต่อวินาที

คำสำคัญ: ระบบปฏิบัติการแอนดรอยด์
แอปพลิเคชัน web scrapping Elasticsearch

บทนำ

ปัจจุบันการซื้อขายสินค้าออนไลน์เริ่มมากขึ้น เพราะความก้าวหน้าทางเทคโนโลยีสมาร์ทโฟน และในช่วงสถานการณ์โควิด การที่ออกไปเลือกซื้อสินค้าตามที่ต่างๆ ทำให้เกิดการเสี่ยงต่อการติดโรค ผู้คนจึงหันมาซื้อสินค้าออนไลน์มากขึ้น ซึ่งการซื้อขายแต่ละครั้ง ผู้สั่งซื้อต้องค้นหาเลือกซื้อสินค้าตามที่ต้องการ อาจจะดูจากราคา โปรโมชั่น คุณภาพสินค้า หรือความน่าเชื่อถือของร้านจากเว็บไซต์ผู้ให้บริการ บางครั้งอาจจะต้องค้นหาจากเว็บไซต์หลายเว็บไซต์ เพื่อเปรียบเทียบราคา ให้ตรงกับความต้องการมากที่สุด

ด้วยเหตุนี้ผู้จัดทำได้พัฒนาแอปพลิเคชันสินค้าลดราคา โดยระบบจะทำการดึงข้อมูลจากหน้าเว็บไซต์และนำสินค้าลดราคามาจัดเก็บไว้ ซึ่งทำให้ผู้ใช้งานเลือกดูสินค้าที่ลดราคาจากเว็บไซต์ต่างๆ ที่ทางระบบได้กำหนดไว้ ทำให้ทราบถึงสินค้าแต่ละเว็บไซต์ว่าแตกต่างกันมากน้อยเพียงใด เพื่อเป็นข้อมูลในการตัดสินใจเลือกซื้อสินค้าลดราคาให้ตรงตามความต้องการ

วิธีการดำเนินงาน

พัฒนาบอท(Bot) เพื่อดึงข้อมูลในเว็บไซต์

เพื่อให้การดึงหน้าเว็บไซต์ทำงานได้เร็วมากขึ้น
ได้แบ่งการทำงานออกเป็นส่วย่อยและพัฒนาบอท
เป็น 3 บอท ได้แก่

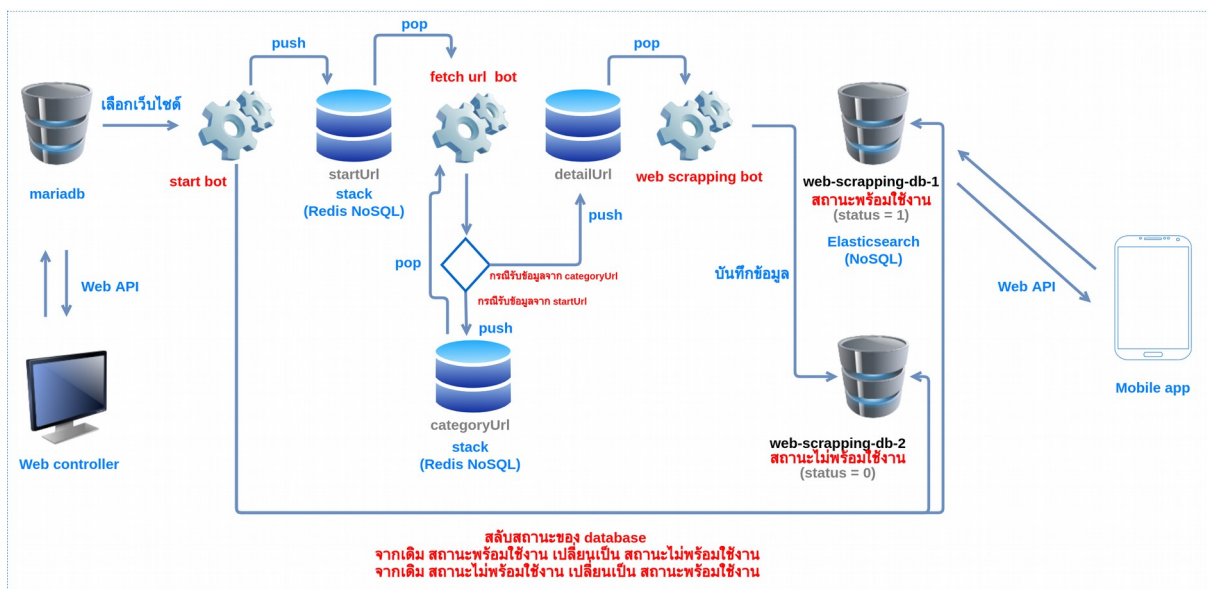
- 1 start bot
- 2 fetch url bot
- 3 web scrapping bot

start bot ทำงานตามเวลาที่ได้ตั้งค่าไว้ในระบบ
จากนั้น start bot ทำการเลือกเว็บไซต์ที่ต้องการดึง
ข้อมูลตามที่ได้ตั้งค่าไว้ในระบบ และทำการจัดเก็บ
ข้อมูลลงใน Redis ที่มีชื่อว่า startUrl เปรียบเสมือน
เป็น stack ซึ่งเป็นลักษณะของ temporary
database เพื่อส่งต่อการทำงานให้ fetch url bot
ทำงานต่อ

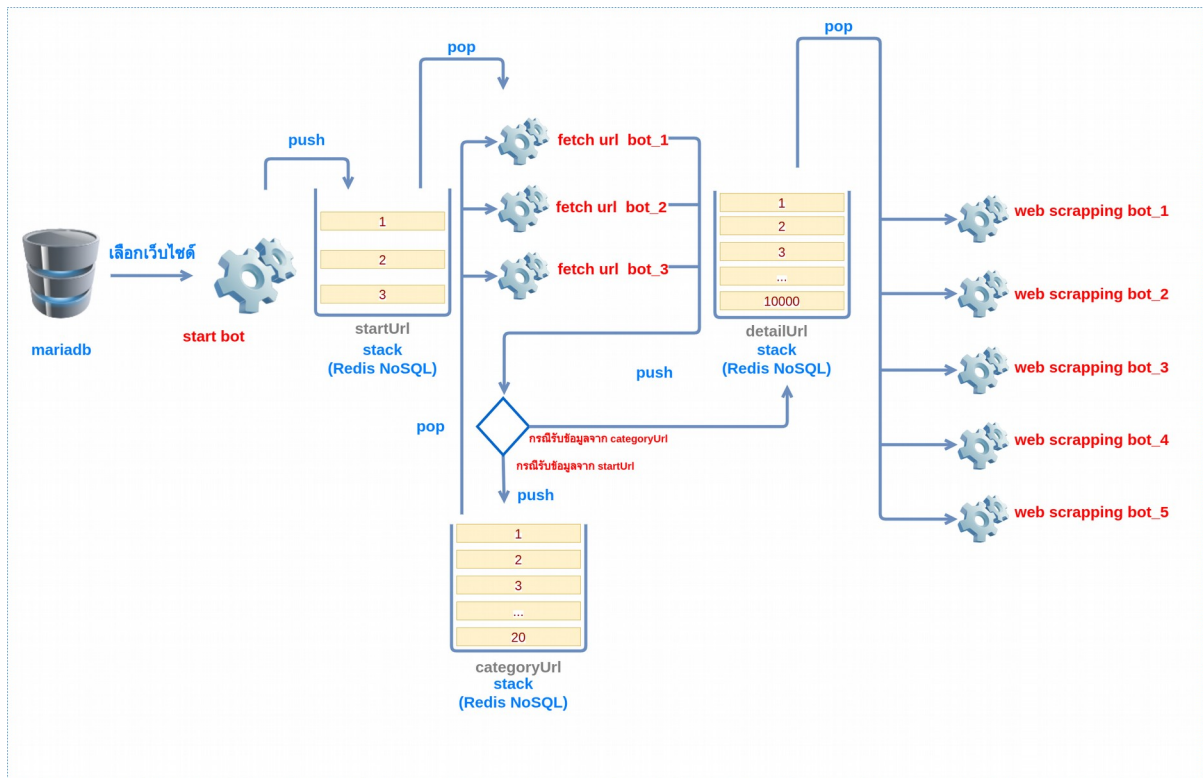
fetch url bot นำข้อมูลที่ได้จาก startUrl และ
ทำงานตามเงื่อนไข จากนั้นลำดับสุดท้ายจะทำการ
จัดเก็บข้อมูลที่ได้ลงใน Redis ที่มีชื่อว่า detailUrl
เพื่อให้ web scrapping bot ทำงานต่อ

web scrapping bot นำข้อมูลที่ได้จาก
detailUrl เรียบร้อยแล้วทำการดึงข้อมูลในเว็บไซต์
จากนั้นนำข้อมูลที่ได้จัดเก็บลง Elasticsearch
database ดังรูปที่ 1

จะเห็นได้ว่าการทำงานดังกล่าวเป็นลักษณะแบบ
ส่งต่อการทำงานเป็นช่วง โดยไม่จำเป็นต้องให้บอท
ตัวใดตัวหนึ่งทำงานจนเสร็จ ถ้ามีข้อมูลใน Redis
database บอทตัวถัดไปสามารถทำงานต่อได้เลย
เป็นลักษณะของ pipeline และเมื่อเกิดปัญหาหาคอ
ขวดคือข้อมูลใน Redis มีจำนวนมากจนบอททำงาน
ไม่ทัน ก็สามารถเพิ่มจำนวนบอทในแต่ละช่วง เพื่อแก้
ปัญหาดังกล่าวได้ ดังรูปที่ 2



รูปที่ 1 ขั้นตอนการทำงานของบอทเพื่อดึงข้อมูลหน้าเว็บไซต์



รูปที่ 2 แสดงการเพิ่มจำนวนบอท

ผลการศึกษาและอภิปรายผล

ผลการพัฒนา Bot เพื่อดึงข้อมูลหน้าเว็บไซต์

1 ทดสอบความครบถ้วนของข้อมูล ได้ทำการทดสอบดึงข้อมูลหน้าเว็บไซต์ เทสโก้ โลตัส ผลปรากฏว่าความครบถ้วนของข้อมูลอยู่ในระดับ 99.95% ดังตารางที่ 1

2 ทดสอบเพิ่มจำนวนบอท เพื่อดูระยะเวลาการดึงข้อมูลทั้ง 3 เว็บไซต์ คือ เทสโก้ โลตัส แม็คโคร บิ๊กซี ผลปรากฏว่าเมื่อเพิ่มจำนวนบอท เป็น 4 บอท เวลาการทำงานลดเกินครึ่ง จากนั้นเพิ่มจำนวนบอทไปตามลำดับ เวลาการทำงานก็ลดลงเรื่อยๆ ดังตารางที่ 2 และรูปที่ 3

ตารางที่ 1 ผลการทดสอบความครบถ้วนของข้อมูล

ลำดับ	วันที่	ข้อมูลสินค้าทั้งหมด	ข้อมูลสินค้าที่ได้	เปอร์เซ็นต์
1	10/05/64	4235	4230	99.88
2	10/05/64	4235	4235	100
3	10/05/64	4235	4235	100
4	10/05/64	4235	4235	100
5	10/05/64	4235	4223	99.71
6	10/05/64	4235	4235	100
7	10/05/64	4235	4234	99.98
8	10/05/64	4235	4235	100
9	10/05/64	4235	4232	99.93
10	10/05/64	4235	4235	100
			ค่าเฉลี่ย	99.95

ตารางที่ 2 ผลการทดสอบเพิ่มจำนวนบอท

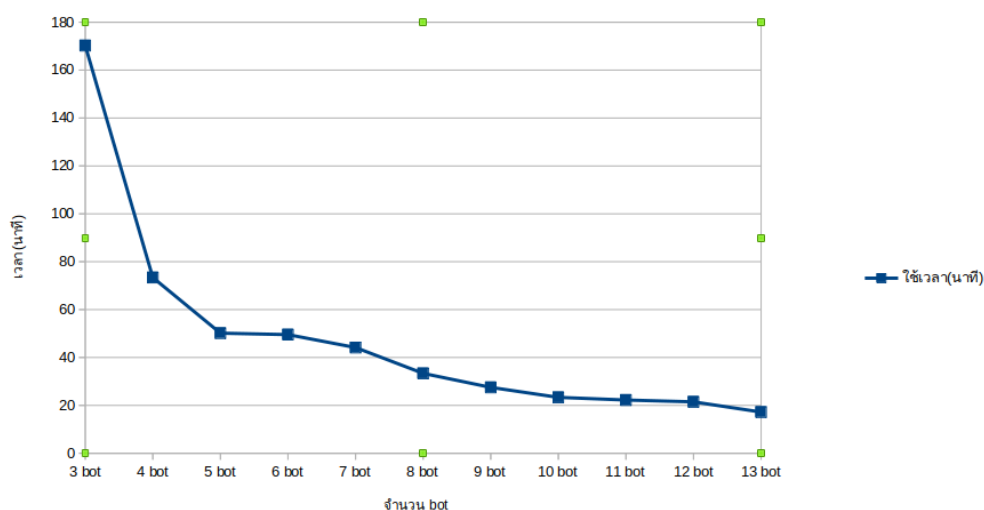
ลำดับ	วันที่	จำนวนบอท	ใช้เวลา (นาท)
1	11/03/64	3 Bot	170.24
2	11/03/64	4 Bot	73.38
3	11/03/64	5 Bot	50.16
4	11/03/64	6 Bot	49.55
5	11/03/64	7 Bot	44.14
6	11/03/64	8 Bot	33.36
7	11/03/64	9 Bot	27.51
8	11/03/64	10 Bot	23.34
9	11/03/64	11 Bot	22.24
10	11/03/64	12 Bot	21.47
11	11/03/64	13 Bot	17.21

ข้อมูลที่ได้แสดงผ่านแอปพลิเคชันค้นหาสินค้าลดราคา



การติดตั้งแอปพลิเคชันค้นหาสินค้าลดราคาสามารถดาวน์โหลดได้จาก Play Store โดยทำการค้นคำว่า “ค้นหาสินค้าลดราคา” จากนั้นกดค้นหาจะพบแอปพลิเคชัน ค้นหาสินค้าลดราคาจากนั้นกดติดตั้ง ดังรูปที่ 4

รูปที่ 4 หน้าจอแสดงการติดตั้งของแอปพลิเคชัน



รูปที่ 3 กราฟแสดงการใช้เวลาทำงานของบอท

แอปพลิเคชันสินค้าราคา ในเมนูค้นหาได้ใช้เครื่องมือของ Elasticsearch ช่วยค้นหา โดยเลือกใช้ฟังก์ชันค้นหาแบบ Regular Expression ซึ่งเป็นการกำหนดรูปแบบหรือกลุ่มคำ เพื่อเอาไว้ใช้ค้นหาข้อความต่างๆตามที่ต้องการ

ทดสอบ 100 คำค้น แยกเป็น 10 หมวดหมู่
หมวดหมู่ละ 10 คำค้น ผลลัพธ์ที่ได้

1. ไม่พบข้อมูล 11 คำค้น
2. พบข้อมูล 89 คำค้น คิดเป็นค่าเฉลี่ย

81.30 % ของการค้นหาพบสินค้าตามที่ต้องการ

ตารางที่ 3 ผลการทดสอบคำค้น

ลำดับ	หมวดหมู่สินค้า	จำนวนเปอร์เซ็นต์ค้นพบสินค้าตามที่ต้องการ
1	เครื่องใช้ไฟฟ้าและอุปกรณ์อิเล็กทรอนิกส์	97.12
2	เครื่องเขียนและอุปกรณ์สำนักงาน	61.90
3	ผลิตภัณฑ์เพื่อสุขภาพความงาม	98.33
4	ผลิตภัณฑ์ทำความสะอาดและของใช้ในครัวเรือน	82.32
5	แม่และเด็ก	97.47
6	เครื่องดื่มและขนมขบเคี้ยว	83.77
7	อาหารแห้ง อาหารกระป๋อง	91.81
8	อาหารสด อาหารแช่แข็ง เบเกอรี่	28.82
9	ผลิตภัณฑ์สำหรับสัตว์เลี้ยง	100
10	เสื้อผ้าและเครื่องแต่งกาย	71.43

ผลการทดสอบเซิร์ฟเวอร์เพื่อดูการรองรับการใช้งานจากผู้ใช้งาน

ปัจจุบันใช้งาน 1 เซิร์ฟเวอร์ (2 cpu ram 4 gb ssd 80 gb) รองรับการใช้งานได้ 270 ผู้ใช้งานต่อวินาที

ตารางที่ 4 ผลการทดสอบ load testing

ลำดับ	จำนวน user ต่อ 1 วินาที	ใช้ cpu server (%)	error (%)	error message
1	20	20.03	0.00	
2	40	28.57	0.00	
3	60	39.18	0.00	
4	80	40.82	0.00	
5	100	47.24	0.00	
6	120	50.51	0.00	
7	140	53.81	0.00	
8	160	57.22	0.00	
9	180	64.4	0.00	
10	200	72.28	0.00	
11	270	77.16	0.00	
12	271	79.8	7.01	Connection timed out

สรุปผล

ผลการทดสอบจากผู้ใช้งาน 10 คน โดยแยกเป็น
ผู้ใช้งานทั่วไป 5 คน และผู้พัฒนาระบบ 5 คน ผล
ปรากฏว่า ผู้ใช้มีความพึงพอใจอยู่ในระดับ “มาก”
($\bar{X} = 4.20$) ส่วนผู้พัฒนาระบบมีความพึงพอใจในม
มองการออกแบบและฟังก์ชันการใช้งานอยู่ในระดับ
“มาก” ($\bar{X} = 4.20$)

ตารางที่ 5 ผลจากการประเมินจากผู้ใช้งาน
แอปพลิเคชัน

แบบประเมิน	ค่าเฉลี่ย	ค่าเบี่ยงเบน
ผู้ใช้งานทั่วไป	4.20	0.59
ผู้พัฒนาระบบ	4.20	0.56

กิตติกรรมประกาศ

การจัดทำโครงงานฉบับนี้สำเร็จได้ด้วยดีต้อง
กราบขอบพระคุณอาจารย์ที่ปรึกษา อาจารย์
ประหยัด เลวัน ที่ได้ให้คำแนะนำ ตรวจสอบโครงงาน
เพื่อปรับปรุงแก้ไขข้อบกพร่องต่างๆ และขอ
ขอบพระคุณคณาจารย์ทุกท่าน ที่ได้ประสิทธิ์ประสาท
วิชาความรู้ เพื่อนำความรู้ที่ได้มาจัดทำโครงงานจน
สำเร็จลุล่วงไปด้วยดี ผู้จัดทำจึงขอขอบพระคุณมา ณ
โอกาสนี้

เอกสารอ้างอิง

1. AWS. (2564). [ออนไลน์] ElasticSearch.
สืบค้นจาก
[https://aws.amazon.com/th/elasticsearch-
service/the-elk-stack/what-is-elasticsearch/](https://aws.amazon.com/th/elasticsearch-service/the-elk-stack/what-is-elasticsearch/)
(วันที่สืบค้น 9 เมษายน 2564)
2. Softmelt. (2554). [ออนไลน์] Redis คืออะไร ?.
สืบค้นจาก
[https://www.softmelt.com/article.php?
id=564](https://www.softmelt.com/article.php?id=564) (วันที่สืบค้น 9 เมษายน 2564)
3. AWS. (2564). [ออนไลน์] Docker คืออะไร.
สืบค้นจาก
<https://aws.amazon.com/th/docker/> (วันที่
สืบค้น 9 เมษายน 2564)
4. STACKPYTHON. (4 ธันวาคม 2563). [ออนไลน์]
Web Scraping. สืบค้นจาก
[https://stackpython.co/tutorial/web-scraping-
python-beautifulsoup-requests](https://stackpython.co/tutorial/web-scraping-python-beautifulsoup-requests) (วันที่สืบค้น 9
เมษายน 2564)
5. Thirawat T. (26 กรกฎาคม 2560). [ออนไลน์]
Regular Expressions คืออะไร ?. สืบค้นจาก
[https://medium.com/@_trw/regular-
expressions-
%E0%B8%84%E0%B8%B7%E0%B8%AD
%E0%B8%AD
%E0%B8%B0%E0%B9%84%E0%B8%A3-
2fab4a91ea34](https://medium.com/@_trw/regular-expressions-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3-2fab4a91ea34)