



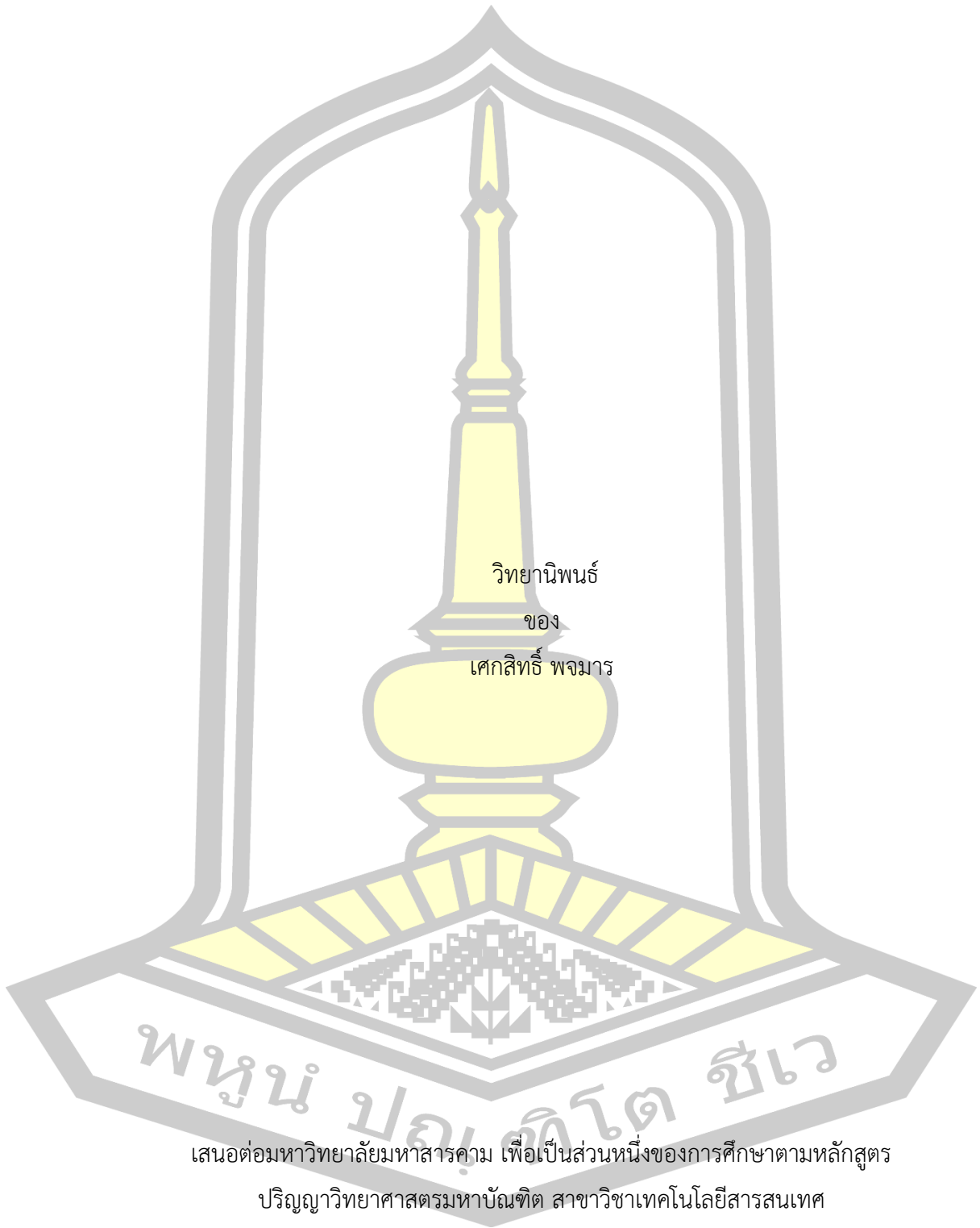
การพัฒนาต้นแบบการบีบอัดแบบไม่สูญเสียข้อมูลบนเว็บแอปพลิเคชัน

วิทยานิพนธ์
ของ
เสกสิทธิ์ พงมาร

เสนอต่อมหาวิทยาลัยมหาสารคาม เพื่อเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
พฤษภาคม 2563

ลิขสิทธิ์เป็นของมหาวิทยาลัยมหาสารคาม

การพัฒนาต้นแบบการบีบอัดแบบไม่สูญเสียข้อมูลบนเว็บแอปพลิเคชัน



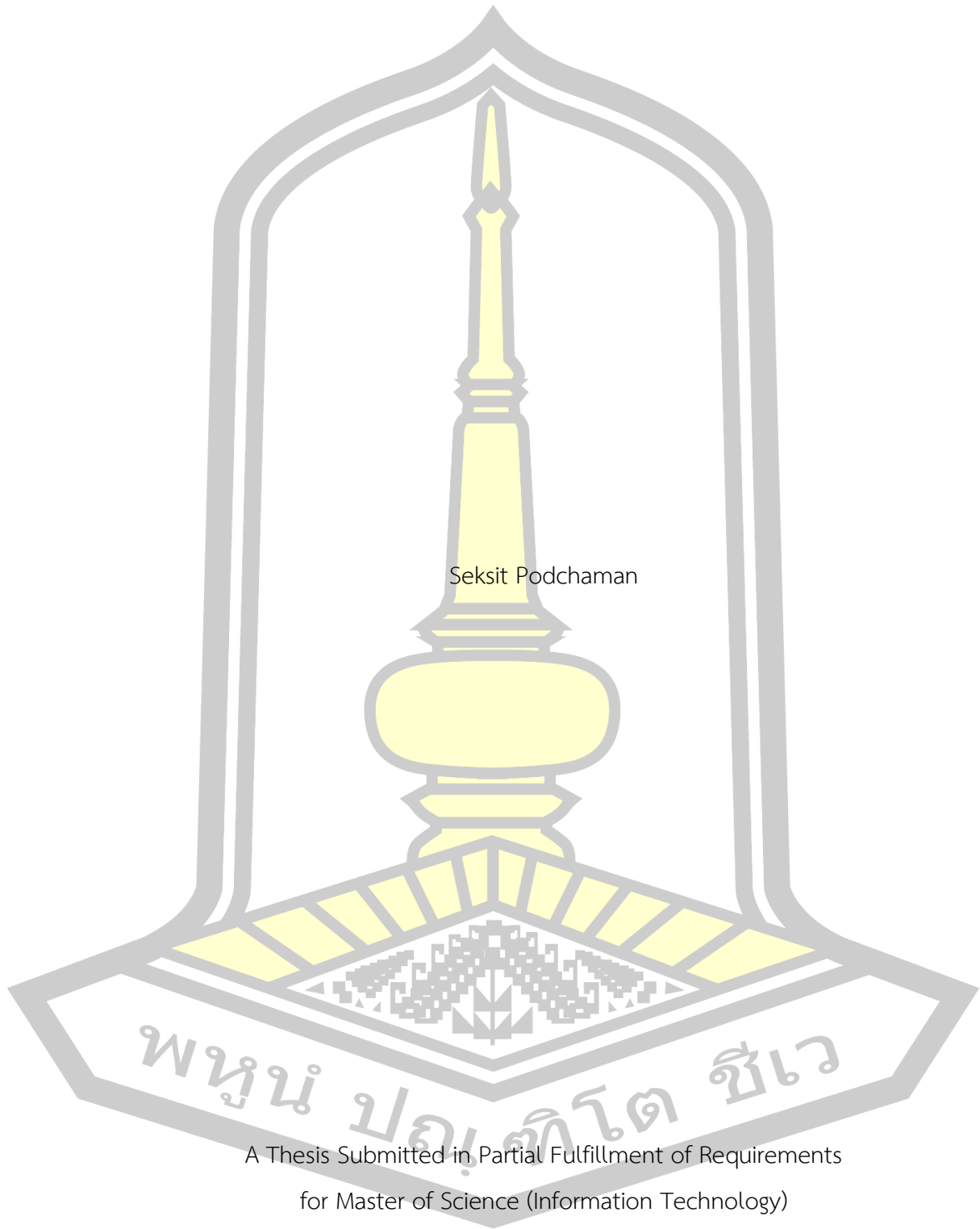
เสนอต่อมหาวิทยาลัยมหาสารคาม เพื่อเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

พฤษภาคม 2563

ลิขสิทธิ์เป็นของมหาวิทยาลัยมหาสารคาม

Developing Prototype of Lossless Data Compression on Web Applications



Seksit Podchaman

A Thesis Submitted in Partial Fulfillment of Requirements
for Master of Science (Information Technology)

May 2020

Copyright of Maharakham University



คณะกรรมการสอบวิทยานิพนธ์ ได้พิจารณาวิทยานิพนธ์ของนายเศกสิทธิ์ พงมาร แล้ว เห็นสมควรรับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชา เทคโนโลยีสารสนเทศ ของมหาวิทยาลัยมหาสารคาม

คณะกรรมการสอบวิทยานิพนธ์

ประธานกรรมการ

(รศ. ดร. สิทธิชัย บุขหมั่น)

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

(ผศ. ดร. จารี ทองคำ)

กรรมการ

(ผศ. ดร. แกมกาญจน์ สมประเสริฐศรี)

กรรมการ

(ดร. สอาทิตย์ แสงประดิษฐ์)

มหาวิทยาลัยอนุมัติให้รับวิทยานิพนธ์ฉบับนี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร ปริญญา วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ ของมหาวิทยาลัยมหาสารคาม

(ผศ. ศศิธร แก้วมั่น)

คณบดีคณะวิทยาการสารสนเทศ

(รศ. ดร. กริสน์ ชัยมูล)

คณบดีบัณฑิตวิทยาลัย

ชื่อเรื่อง	การพัฒนาต้นแบบการบีบอัดแบบไม่สูญเสียข้อมูลบนเว็บแอปพลิเคชัน		
ผู้วิจัย	เศกสิทธิ์ พงมาร		
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร. จาริ ทองคำ		
ปริญญา	วิทยาศาสตรมหาบัณฑิต	สาขาวิชา	เทคโนโลยีสารสนเทศ
มหาวิทยาลัย	มหาวิทยาลัยมหาสารคาม	ปีที่พิมพ์	2563

บทคัดย่อ

งานวิจัยนี้มีวัตถุประสงค์เพื่อเปรียบเทียบอัลกอริทึมการบีบอัดแบบไม่สูญเสียข้อมูลเพื่อลดขนาดไฟล์ข้อมูลให้เล็กลงก่อนทำการส่งไฟล์ผ่านเครือข่ายอินเทอร์เน็ต และใช้เทคนิคการคลายข้อมูลที่เครื่องผู้ใช้ ซึ่งจะทำให้การส่งข้อมูลผ่านเว็บแอปพลิเคชันมีประสิทธิภาพและมีความรวดเร็วเพิ่มขึ้น กลุ่มไฟล์ข้อมูล 3 กลุ่มได้ถูกนำมาใช้ในการทดลองคือ กลุ่มไฟล์เอกสาร กลุ่มไฟล์รูปภาพ และกลุ่มไฟล์เสียงและวิดีโอ จำนวน 150 ไฟล์ โดยใช้อัลกอริทึมการบีบอัดแบบไม่สูญเสียข้อมูลจำนวน 5 อัลกอริทึม ได้แก่ Huffman Coding, Deflate, BZip2, LZMA และ LZ4 ในการวัดประสิทธิภาพการบีบอัดข้อมูล ผู้วิจัยได้ใช้อัตราความเร็วในการบีบอัด อัตราความเร็วในการคลายข้อมูล อัตราส่วนการบีบอัด และเวลารวมทุกกระบวนการ จากนั้นนำผลการเปรียบเทียบที่มีประสิทธิภาพไปสร้างต้นแบบการบีบอัดข้อมูล จากการทดลองพบว่าอัลกอริทึม LZ4 มีประสิทธิภาพในการบีบอัดข้อมูลด้วยอัตราความเร็วรวมทุกกระบวนการทำงานดีที่สุดที่อัตราเฉลี่ย 7.3806 ไรต์ต่อวินาที

คำสำคัญ : การบีบอัดข้อมูล, อัลกอริทึมการบีบอัดแบบไม่สูญเสียข้อมูล, อัลกอริทึม LZ4

พหุณฺ์ ปณฺุ ทฺิโต ชฺีเว

TITLE	Developing Prototype of Lossless Data Compression on Web Applications		
AUTHOR	Seksit Podchaman		
ADVISORS	Assistant Professor Jaree Thongkam , Ph.D.		
DEGREE	Master of Science	MAJOR	Information Technology
UNIVERSITY	Maharakham University	YEAR	2020

ABSTRACT

The paper aims to compare different lossless data compression algorithms that are used to reduce the size of data before transmitting it over the Internet. Because the data is preserved, it can be decompressed and restored to its original state. This facilitates fast and efficient data transfer in web applications. In this research, a total of 150 files of 3 different file types are used including text files, image files and audio/video files. 5 lossless data compression algorithms including Huffman Coding, Deflate, BZip2, LZMA and LZ4 are studied and compared in terms of compression/decompression speed compression rate and total processing time, then use the effective comparison results to create the data compression prototype. The results show that LZ4 algorithm produces the best overall performance, with the average rate of 7.3806 bytes per second (bit/s).

Keyword : Data Compression, Lossless Data Compression Algorithms, LZ4 Algorithms

พหุ ม ประทีป ชีวะ

กิตติกรรมประกาศ

งานวิจัยนี้ สำเร็จลุล่วงได้ด้วยความสามารถอย่างสูงจาก ผศ.ดร.จारी ทองคำ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่กรุณาให้คำแนะนำ ปรึกษา รวมทั้งเสนอแนะแนวทางในการดำเนินงานวิจัย ด้วยความเอาใจใส่อย่างดียิ่ง ซึ่งเป็นประโยชน์ต่อผู้วิจัยอย่างยิ่ง

เศกสิทธิ์ พจมาร

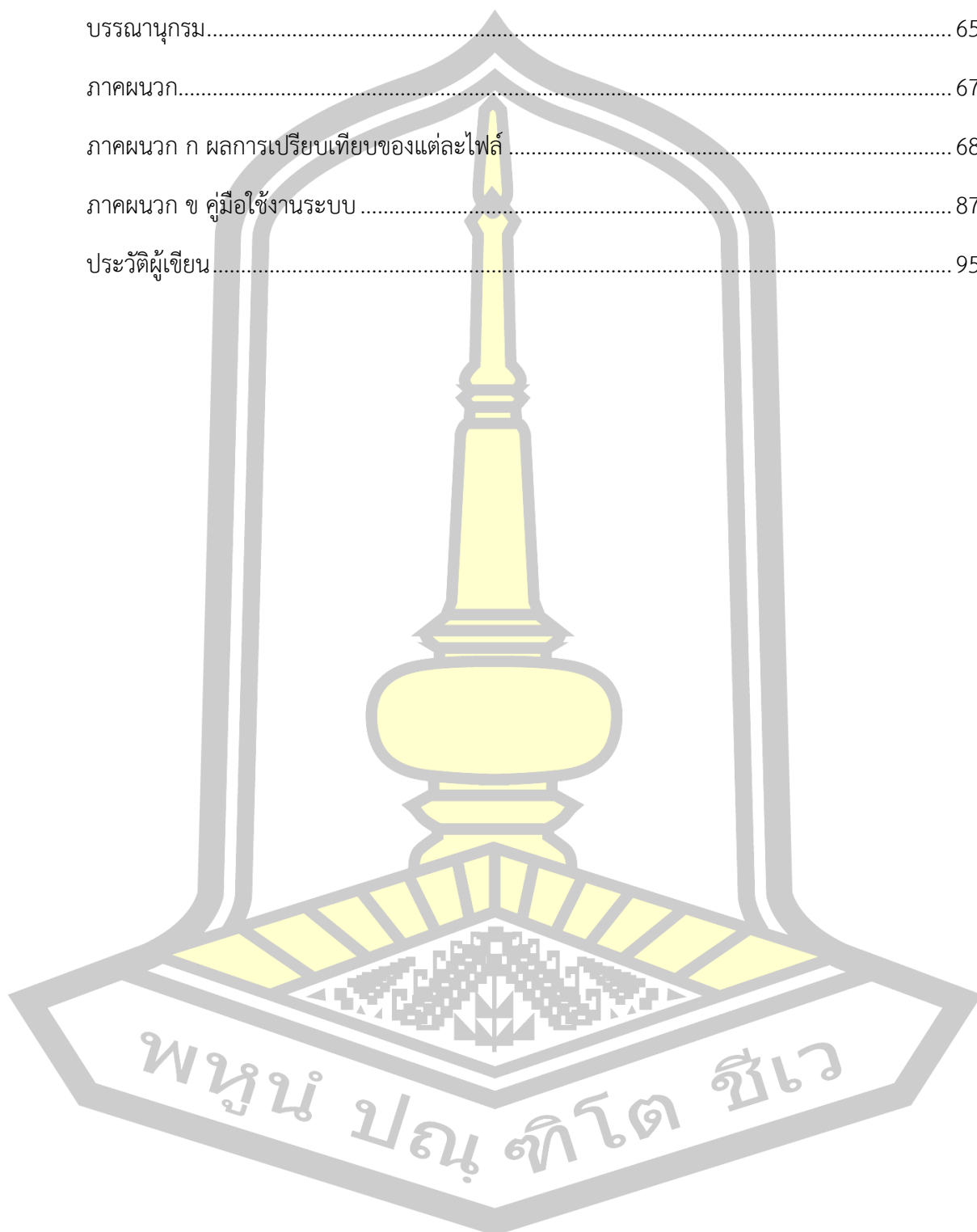


สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ.....	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ.....	ช
สารบัญตาราง.....	ญ
สารบัญภาพ.....	ฎ
บทที่ 1 บทนำ.....	1
1.1 หลักการและเหตุผล.....	1
1.2 วัตถุประสงค์ของการวิจัย.....	2
1.3 ขอบเขตของการวิจัย.....	2
1.4 ความสำคัญของการวิจัย.....	4
1.5 นิยามศัพท์เฉพาะ.....	5
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	6
2.1 ทฤษฎีที่เกี่ยวข้อง.....	6
2.1.1 การบีบอัดข้อมูล.....	6
2.1.2 เทคนิคฮัฟแมน โค้ด (Huffman Codes).....	9
2.1.3 เทคนิคแอลแซสเจ็ดเจ็ด (LZ77).....	14
2.1.4 เทคนิคบีซีพีทู (BZip2).....	15
2.1.5 เทคนิคดีเฟ้ค (DEFLATE).....	17
2.1.6 เทคนิคแอลแซสโฟร์ (LZ4).....	18
2.1.7 เทคนิคแอลแซสเอ็มเอ (LZMA).....	20

2.2 งานวิจัยที่เกี่ยวข้อง.....	21
บทที่ 3 วิธีดำเนินการวิจัย.....	25
3.1 การเก็บรวบรวมไฟล์ข้อมูลและจัดการแยกไฟล์ข้อมูลตามหมวดหมู่ขนาดไฟล์.....	25
3.1.1 หมวดไฟล์เอกสาร.....	26
3.1.2 หมวดไฟล์รูปภาพ.....	27
3.1.3 หมวดไฟล์เพลงและวิดีโอ.....	28
3.2 ประมวลผลการบีบอัด/คลายข้อมูล และการรับ-ส่งข้อมูล.....	29
3.2.1 กระบวนการบีบอัดข้อมูล.....	30
3.2.2 กระบวนการคลายข้อมูล.....	31
3.2.3 กระบวนการรับ-ส่งข้อมูลผ่านเว็บแอปพลิเคชัน.....	32
3.2.4 การตรวจสอบความถูกต้องของไฟล์หลังการคลายข้อมูล.....	32
3.3 การประเมินประสิทธิภาพการทำงาน.....	34
บทที่ 4 ผลการดำเนินงานวิจัย.....	38
4.1 ผลการเปรียบเทียบอัตราการความเร็วการบีบอัด (Compression speed).....	38
4.2 ผลการเปรียบเทียบอัตราการความเร็วการคลายข้อมูล (Decompression speed).....	40
4.3 ผลการเปรียบเทียบอัตราส่วนการบีบอัดข้อมูล (Compression Ratio).....	43
4.4 ผลการเปรียบเทียบประสิทธิภาพเวลารวมทุกกระบวนการ (Overall Time).....	45
4.5 ผลการทดสอบประสิทธิภาพการบีบอัด/การคลายข้อมูล ตามช่วงกลุ่มขนาดไฟล์.....	49
4.6 ผลการทดสอบประสิทธิภาพการบีบอัดด้วยสถิติเชิงอนุมาน.....	50
4.7 ผลการทดสอบต้นแบบการบีบอัดข้อมูล.....	57
บทที่ 5 สรุปผลและข้อเสนอแนะ.....	63
5.1 สรุปผลการดำเนินงาน.....	63
5.2 อภิปรายผล.....	63
5.3 ปัญหาและอุปสรรค.....	64

5.4 ข้อเสนอแนะสำหรับการวิจัย	64
บรรณานุกรม.....	65
ภาคผนวก.....	67
ภาคผนวก ก ผลการเปรียบเทียบของแต่ละไฟล์	68
ภาคผนวก ข คู่มือใช้งานระบบ	87
ประวัติผู้เขียน.....	95

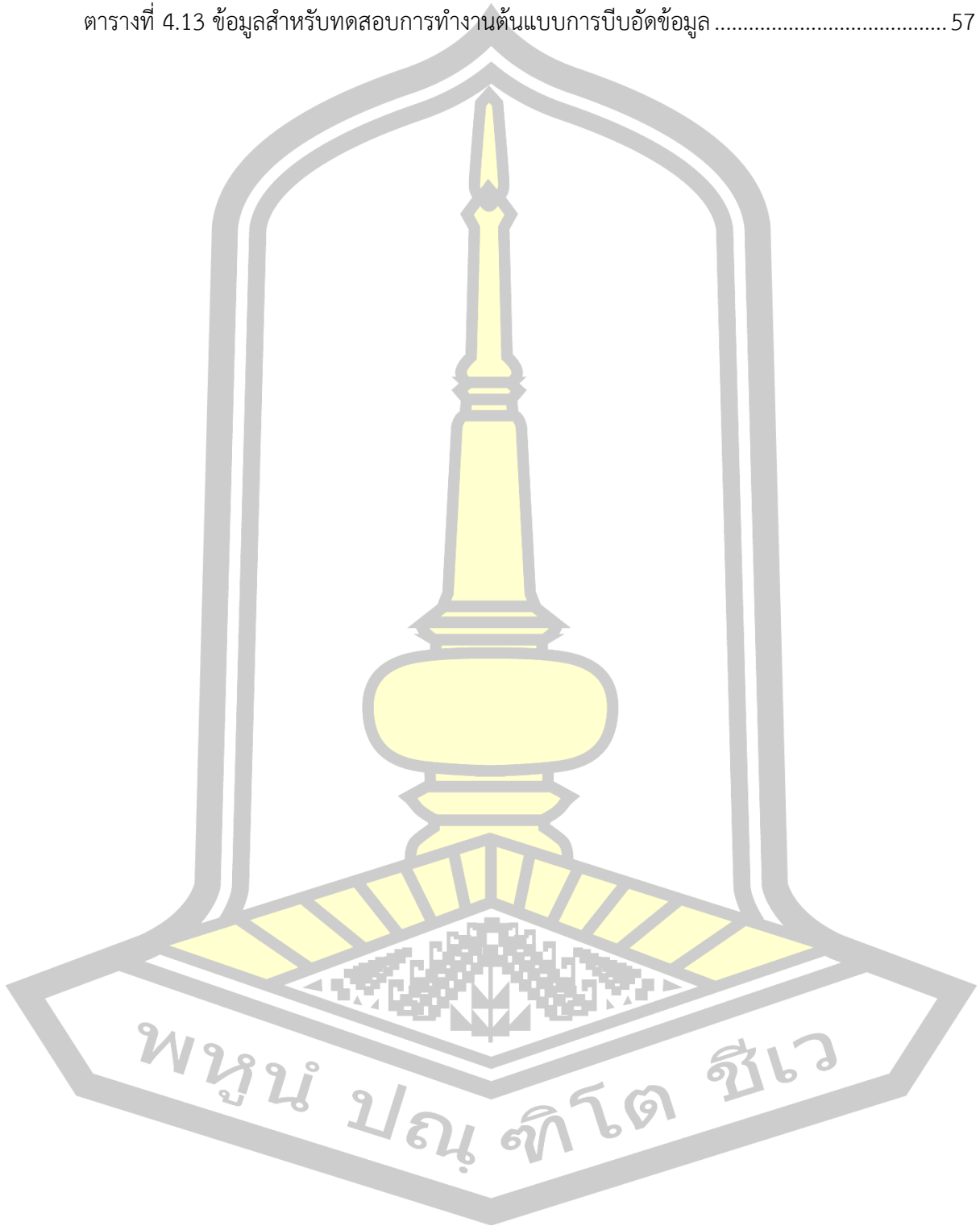


สารบัญตาราง

หน้า

ตารางที่ 1.1 การแบ่งหมวดไฟล์ข้อมูล.....	2
ตารางที่ 1.2 ข้อมูลประเภทไฟล์ 15 ช่วงขนาดไฟล์ข้อมูล.....	3
ตารางที่ 2.1 สัญลักษณ์และจำนวนของการปรากฏของข้อมูลสัญลักษณ์.....	9
ตารางที่ 2.2 แสดงรหัสแทนสัญลักษณ์ของการเข้ารหัสแบบฮัฟแมน.....	13
ตารางที่ 2.3 การใช้งานของตาราง Symbol ในอัลกอริทึม DEFLATE.....	18
ตารางที่ 2.4 ผลการวัดประสิทธิภาพของ LZ4 เมื่อเทียบกับเครื่องมือในการบีบอัดอื่น ๆ.....	19
ตารางที่ 2.5 คำอธิบายแพ็คเกจเกิดของ LZMA.....	21
ตารางที่ 2.6 รายละเอียดรหัสความยาวข้อมูล.....	21
ตารางที่ 3.1 ข้อมูลหมวดไฟล์เอกสาร.....	26
ตารางที่ 3.2 ข้อมูลหมวดไฟล์รูปภาพ.....	27
ตารางที่ 3.3 ข้อมูลหมวดไฟล์เพลงและวิดีโอ.....	28
ตารางที่ 4.1 อัตราความเร็วในการบีบอัดข้อมูล.....	38
ตารางที่ 4.2 อัตราความเร็วในการบีบอัดข้อมูล แยกตามช่วงขนาดไฟล์ข้อมูล.....	39
ตารางที่ 4.3 อัตราความเร็วในการคลายข้อมูล.....	41
ตารางที่ 4.4 อัตราความเร็วในการคลายข้อมูล แยกตามช่วงขนาดไฟล์ข้อมูล.....	41
ตารางที่ 4.5 ร้อยละอัตราส่วนการบีบอัดข้อมูล.....	43
ตารางที่ 4.6 ร้อยละอัตราส่วนการบีบอัดข้อมูลแยกตามช่วงขนาดไฟล์ข้อมูล.....	44
ตารางที่ 4.7 อัตราความเร็วรวมทุกกระบวนการ แยกตามช่วงขนาดไฟล์ข้อมูล.....	47
ตารางที่ 4.8 อัตราความเร็วในการรับ-ส่งข้อมูล แยกตามช่วงขนาดไฟล์ข้อมูล.....	48
ตารางที่ 4.9 ผลการเปรียบเทียบประสิทธิภาพ ตามช่วงกลุ่มขนาดไฟล์.....	50
ตารางที่ 4.10 ผลการทดสอบหาความแตกต่างค่าที (t-test) ตามหมวดไฟล์ข้อมูล.....	51
ตารางที่ 4.11 ผลการทดสอบหาความแตกต่างค่าที (t-test) หมวดไฟล์รูปภาพ.....	53

ตารางที่ 4.12 ผลการทดสอบหาความแตกต่างค่าที (t-test) ตามหมวดไฟล์เสียงและวิดีโอ.....	55
ตารางที่ 4.13 ข้อมูลสำหรับทดสอบการทำงานต้นแบบการบีบอัดข้อมูล	57

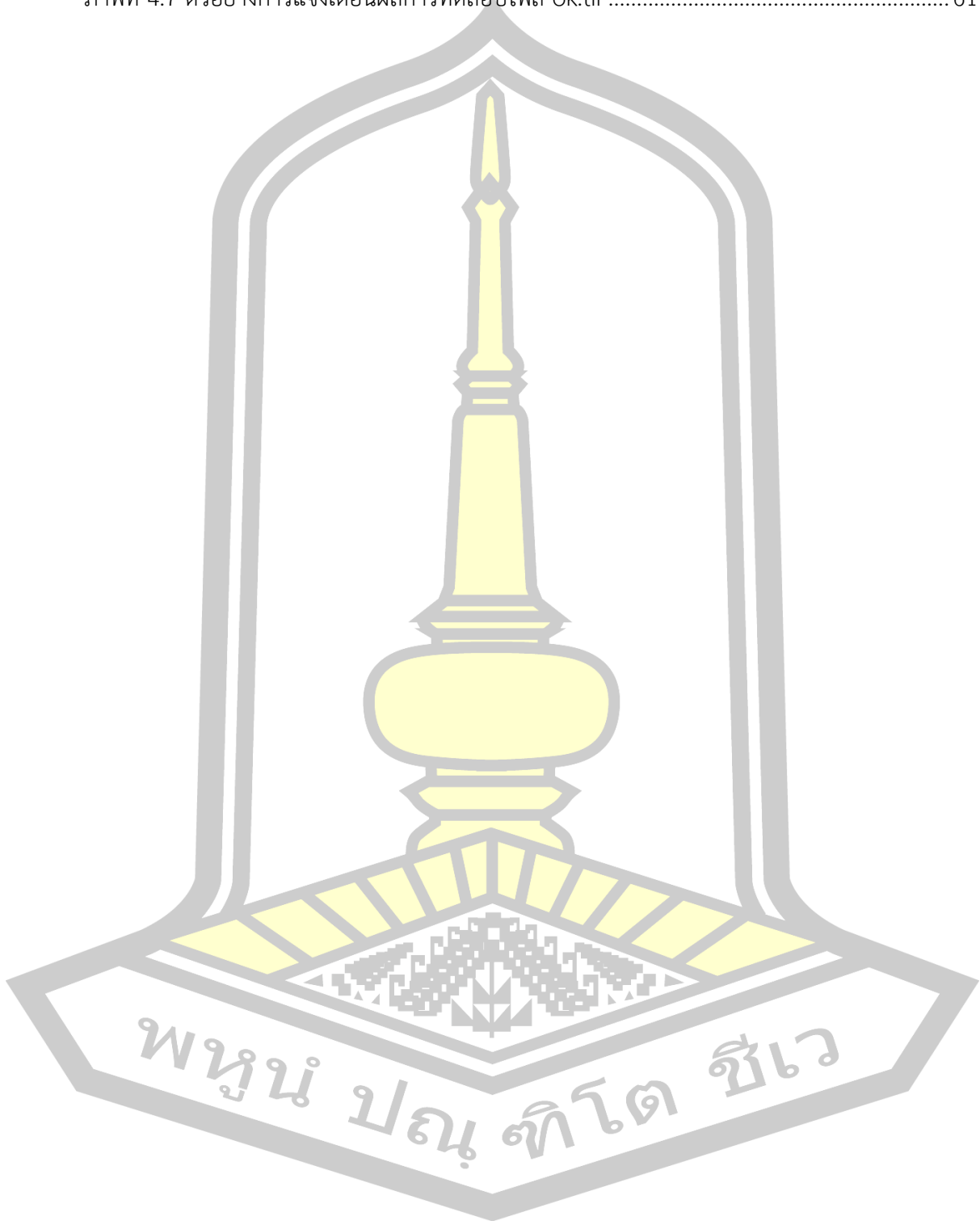


สารบัญภาพ

หน้า

ภาพที่ 2.1	รูปแบบการบีบอัดข้อมูล	6
ภาพที่ 2.2	ข้อมูลกลไกการรหัสด้วย CODEC	8
ภาพที่ 2.3	Flows การบีบอัดข้อมูล/คลายข้อมูล.....	8
ภาพที่ 2.4	ตัวอย่างโครงสร้างโปรแกรมการเข้ารหัสแบบฮัฟแมน (Huffman Pseudocode)	13
ภาพที่ 2.5	ตัวอย่างหน้าต่างของ sliding window	14
ภาพที่ 2.6	แสดงการบีบอัดข้อมูลด้วยวิธีบีบอัด.....	16
ภาพที่ 2.7	แสดงการบีบอัดข้อมูลแบบขนานด้วยวิธีบีบอัด.....	17
ภาพที่ 2.8	ส่วนประกอบของบล็อกการบีบอัดข้อมูล (Compression block) ของ LZ4	19
ภาพที่ 2.9	ผลการเปรียบเทียบการบีบอัดข้อมูลทั้ง 2 หมวดหมู่.....	22
ภาพที่ 3.1	ภาพรวมการประมวลผลการบีบอัด/คลายข้อมูล.....	29
ภาพที่ 3.2	การแบ่งข้อมูลออกเป็นบล็อก.....	29
ภาพที่ 3.3	กระบวนการบีบอัดข้อมูล.....	30
ภาพที่ 3.4	กระบวนการคลายข้อมูล.....	31
ภาพที่ 3.5	กระบวนการรับ-ส่งไฟล์ และการคลายข้อมูล.....	32
ภาพที่ 3.6	ขั้นตอนการอัปเดตและบีบอัดไฟล์.....	35
ภาพที่ 3.7	ขั้นตอนการดาวน์โหลดและคลายข้อมูล.....	36
ภาพที่ 4.1	อัตราความเร็วทุกกระบวนการตั้งแต่เริ่มบีบอัดข้อมูลถึงกระบวนการคลายข้อมูล.....	46
ภาพที่ 4.2	ขั้นตอนในการอัปเดตไฟล์เข้าระบบและทำการบีบอัดข้อมูล.....	58
ภาพที่ 4.3	ตัวอย่างหน้าจอต้นแบบการบีบอัดข้อมูล.....	59
ภาพที่ 4.4	ตัวอย่างการแจ้งเตือนผลการทดสอบไฟล์ Propersal.doc	60
ภาพที่ 4.5	ตัวอย่างการแจ้งเตือนผลการทดสอบไฟล์ KTB656.JPG	60
ภาพที่ 4.6	ตัวอย่างการแจ้งเตือนผลการทดสอบไฟล์ ขอโอกาสแทนเด้อ - บอย พนมไพร.mp3	61

ภาพที่ 4.7 ตัวอย่างการแจ้งเตือนผลการทดสอบไฟล์ ok.tif 61



บทที่ 1

บทนำ

1.1 หลักการและเหตุผล

ปัจจุบันเทคโนโลยีเว็บแอปพลิเคชันได้เข้ามามีบทบาทต่อชีวิตประจำวันมากขึ้น ซึ่งเป็นผลมาจากที่เทคโนโลยีเว็บแอปพลิเคชันได้มีการพัฒนาและเติบโตขึ้นอย่างรวดเร็ว ทำให้การรับ-ส่งข้อมูลมีปริมาณเพิ่มขึ้น [1] ส่งผลให้เครื่องบริการเว็บ (Web Server) มีความเร็วลดลงไม่สามารถให้บริการได้อย่างมีประสิทธิภาพ หรือบางครั้งก็ไม่สามารถเข้าใช้งานระบบงานได้ เนื่องมาจากเครื่องบริการเว็บต้องใช้หน่วยความจำ (Memory) ซีพียู (CPU) พื้นที่เก็บข้อมูล (Disk) และแบนด์วิดท์ (bandwidth) มากเกินไปทำให้ประสิทธิภาพลดลงไป ปัจจัยหลักอีกประการหนึ่งคือปริมาณ หรือขนาดของไฟล์ข้อมูลที่ถูกส่งผ่านเว็บแอปพลิเคชัน ก็มีผลโดยตรงต่อความเร็วในส่งข้อมูล เมื่อไฟล์ข้อมูลมีขนาดใหญ่ขึ้น เวลาในการรับ-ส่งก็ใช้เวลามากขึ้นตามไปด้วย [2] ซึ่งในปัจจุบันมีวิธีการต่าง ๆ ในการเพิ่มประสิทธิภาพให้การรับ-ส่งข้อมูลเพื่อเพิ่มความเร็วในการส่ง วิธีหนึ่งที่ถูกนำมาใช้ในการแก้ปัญหาเรื่องขนาดไฟล์ คือการบีบอัดไฟล์ให้มีขนาดเล็กลงก่อนส่งไฟล์ เพื่อให้มีความเหมาะสมต่อการจัดเก็บไฟล์ข้อมูล และการส่งข้อมูลผ่านเว็บแอปพลิเคชัน

การบีบอัดข้อมูลเป็นที่นิยมในปัจจุบันซึ่งมีนักวิจัยหลายท่านได้พัฒนาอัลกอริทึม เพื่อนำมาใช้ในการบีบอัดข้อมูล Uthayakumar และคณะ [3] ได้นำวิธีการบีบอัดข้อมูลมาใช้ในเครือข่ายเซนเซอร์ไร้สาย (wireless sensor networks : WSN) เนื่องจากโหนดเซนเซอร์ใช้พลังงานจากแบตเตอรี่ที่มีความจุจำกัด การส่งข้อมูลเป็นกระบวนการที่ต้องใช้พลังงานหลักใน WSN จึงมีการนำเทคนิคการประหยัดพลังงานหลาย ๆ รูปแบบมาใช้ และรูปแบบการบีบอัดข้อมูลเป็นเทคนิคที่ใช้พลังงานอย่างมีประสิทธิภาพ ซึ่งช่วยลดปริมาณข้อมูลที่จะส่งในเครือข่ายทำให้ประหยัดพลังงานได้มาก จากการศึกษาทดลองการนำวิธีการบีบอัดข้อมูลมาใช้สามารถประหยัดพลังงานได้มากถึง 87.57% Xudong และ Yiran [4] ได้วิจัยเรื่องการบีบอัดแบบไม่สูญเสียข้อมูล โดยใช้การบีบอัดข้อมูลในรูปแบบของไฟล์ JSON สาเหตุมาจากการแลกเปลี่ยนข้อมูลเดิมในระบบอินเทอร์เน็ตในสรรพสิ่ง (Internet of Things : IOT) ซึ่งมีการแลกเปลี่ยนข้อมูลจากฝั่งไคลเอนต์และฝั่งเซิร์ฟเวอร์เดิมเป็นแบบ xml ซึ่งมีขนาดใหญ่ มาเป็นแบบ JSON พร้อมกับการบีบอัดข้อมูลแบบไม่สูญเสียข้อมูล จำนวน 4 อัลกอริทึมดังนี้ LZMA, GZIP, BZIP และ A-LZMA จากผลการวิจัย อัลกอริทึม BZIP มีประสิทธิภาพที่ดีกว่าวิธีอื่น ๆ Uthayakumar และ Vengattaraman [5] ได้ศึกษาอัลกอริทึมเพื่อเปรียบเทียบการบีบอัดรูปภาพ

จำนวน 71 รูปภาพ และแต่ละภาพมีความบิดเบือนจากความเป็นจริงสูง และทำการเลือกอัลกอริทึมมาทดสอบ จำนวน 6 อัลกอริทึมดังนี้ LZMA, PPM, BWT, LZW, Deflate, LZ77 และ Deflate64 จากผลการวิจัย อัลกอริทึม PPM, LZW และ Deflate64 ได้ผลการบีบอัดที่ดีกว่าวิธีอื่น ๆ

ดังนั้นผู้วิจัย จึงได้มีแนวคิดในการแก้ปัญหาเหล่านี้ ด้วยการนำวิธีการบีบอัดข้อมูลมาใช้ในการลดขนาดของข้อมูลให้เล็กลงก่อนที่จะส่ง และในการวิจัยในครั้งนี้ได้มุ่งเน้นการเปรียบเทียบประสิทธิภาพการบีบอัดแบบไม่สูญเสียข้อมูลจำนวน 5 อัลกอริทึม คือ Huffman Coding, LZMA, LZ4, DEFLATE และ BZip2 โดยการเปรียบเทียบอัตราเวลาความเร็วในการบีบอัด/คลายข้อมูล อัตราส่วนการบีบอัด และความเร็วในการรับ-ส่งไฟล์ข้อมูลผ่านเว็บแอปพลิเคชัน เพื่อทำการเปรียบเทียบประสิทธิภาพการทำงานแต่ละอัลกอริทึมเพื่อให้ได้ผลการบีบอัดที่เหมาะสมกับหมวดหมู่ไฟล์ข้อมูลและประเภทไฟล์ข้อมูล และนำผลการทดสอบที่มีประสิทธิภาพโดยรวมที่ดีที่สุด ไปพัฒนาต้นแบบการบีบอัดข้อมูลต่อไป

1.2 วัตถุประสงค์ของการวิจัย

1.2.1 เพื่อเปรียบเทียบประสิทธิภาพด้านความเร็วของอัลกอริทึมบีบอัดแบบไม่สูญเสียข้อมูล เพื่อให้มีความเหมาะสมกับ หมวดไฟล์เอกสาร หมวดไฟล์รูปภาพ และหมวดไฟล์เพลงและวิดีโอ

1.2.2 เพื่อพัฒนาต้นแบบการบีบอัดข้อมูลผ่านเว็บแอปพลิเคชัน จากผลการเปรียบเทียบประสิทธิภาพการบีบอัดข้อมูล

1.3 ขอบเขตของการวิจัย

1.3.1 ในวิจัยครั้งนี้ได้รวบรวมไฟล์ข้อมูลทั้งหมด จำนวน 150 ไฟล์ข้อมูล ด้วยการแบ่งออกเป็น 3 หมวดหมู่ คือ หมวดไฟล์เอกสาร หมวดไฟล์รูปภาพ และ หมวดไฟล์เพลงและวิดีโอ ดังตารางที่ 1.1

ตารางที่ 1.1 การแบ่งหมวดไฟล์ข้อมูล

ลำดับที่	หมวดไฟล์เอกสาร	หมวดไฟล์รูปภาพ	หมวดไฟล์เพลงและวิดีโอ
1	PDF	JPG	MP3
2	DOC, DOCX	GIF	WAV
3	XLS, XLSX	TIF, TIFF	AVI
4	LOG	PNG	MP4
5	TXT	BMP	3GP
6	BAK	PSD	WMA

จากนั้นทำการแบ่งประเภทของไฟล์ด้วยขนาดที่แตกต่างกัน 15 ขนาดไฟล์ (ประมาณ 1 กิโลไบต์ ถึง 500 เมกะไบต์) การแบ่งประเภทไฟล์ข้อมูลเพื่อให้สามารถทดสอบประสิทธิภาพการทำงานของอัลกอริทึมแต่ละแบบได้อย่างมีประสิทธิภาพ ดังตารางที่ 1.2

ตารางที่ 1.2 ข้อมูลประเภทไฟล์ 15 ช่วงขนาดไฟล์ข้อมูล

ลำดับที่	ช่วงขนาดไฟล์ข้อมูล	จำนวนไฟล์ที่ใช้ในการวิจัย
1	ตั้งแต่ 1KB – 5KB	6
2	ตั้งแต่ 5KB – 10KB	6
3	ตั้งแต่ 10KB – 50KB	5
4	ตั้งแต่ 50KB – 200KB	6
5	ตั้งแต่ 200KB – 500KB	5
6	ตั้งแต่ 500KB – 800KB	6
7	ตั้งแต่ 800KB – 1000KB	8
8	ตั้งแต่ 1MB – 5MB	21
9	ตั้งแต่ 5MB – 10MB	21
10	ตั้งแต่ 10MB – 50MB	19
11	ตั้งแต่ 50MB – 60MB	11
12	ตั้งแต่ 60MB – 70MB	9
13	ตั้งแต่ 70MB – 80MB	9
14	ตั้งแต่ 80MB – 100MB	9
15	ตั้งแต่ 100MB ขึ้นไป	9
รวม		150

1.3.2 พัฒนาระบบด้วยภาษา ASP.NET Core MVC 2.2 เพื่อเปรียบเทียบวิธีการบีบอัดข้อมูลแบบไม่สูญเสีย จำนวน 5 เทคนิค ได้แก่ เทคนิคฮัฟแมน โค้ด (Huffman Codes), เทคนิคบีซีพีทู (BZip2), เทคนิคดีเฟลท (DEFLATE), เทคนิคแอลแซสเอ็มเอ (LZMA) และเทคนิคแอลแซสโฟร์ (LZ4)

1.3.3 พัฒนาด้านแบบการบีบอัดข้อมูล โดยต้นแบบการบีบอัดจะทำการประมวลผลการบีบอัดข้อมูลอัตโนมัติตามผลการเปรียบเทียบประสิทธิภาพการบีบอัดที่ดีที่สุดของแต่ละหมวดไฟล์และช่วงประเภทไฟล์ข้อมูล

1.3.4 พัฒนาโปรแกรมที่ทำหน้าที่ในการดาวน์โหลดไฟล์และคลายข้อมูล (API) ซึ่งจะทำงานบนเครื่องผู้ใช้งาน (Client) เพื่อทำงานร่วมกับ Web Server โดยการรับค่า Parameter ไฟล์ที่ต้องการ

ดาวน์โหลด เมื่อดาวน์โหลดไฟล์ลงมายังที่เครื่อง Client เสร็จเรียบร้อยแล้ว โปรแกรมจะทำการคลายการบีบอัดข้อมูลให้อัตโนมัติ

1.3.5 การบีบอัดข้อมูล (Compression) จะประมวลผลการบีบอัดข้อมูลที่เครื่องบริการเว็บ (Web Server) โดยมีรายละเอียดคุณสมบัติเครื่องที่ใช้ในการทดสอบ ดังนี้

- ฮาร์ดแวร์
 - ซีพียู Intel Core i5-2410M @ 2.30 GHz
 - แรม 8 GB
 - ฮาร์ดดิสก์ขนาด 500 GB
- ซอฟต์แวร์
 - ระบบปฏิบัติการ : Windows 10 Professional 64 Bit
 - โปรแกรม Visual Studio.Net 2017 ด้วยภาษา ASP.NET Core MVC 2.2
 - Internet Information Services (IIS) 7.5
 - Browser Chrome version 14 ขึ้นไป

1.3.6 การคลายการบีบอัดข้อมูล (Decompression) จะประมวลผลการคลายข้อมูลที่เครื่องผู้ใช้งาน (Client) โดยมีรายละเอียดคุณสมบัติเครื่องที่ใช้ในการทดสอบ ดังนี้

- ฮาร์ดแวร์
 - ซีพียู Intel Core i3-2410M @ 2.30 GHz
 - แรม 4 GB
 - ฮาร์ดดิสก์ขนาด 500 GB
- ซอฟต์แวร์
 - ระบบปฏิบัติการ : Windows 10 Professional 32 Bit
 - โปรแกรมสำหรับโหลดไฟล์และการคลายข้อมูล (API) พัฒนาด้วย Visual Studio.Net 2017 ด้วยภาษา C# (Windows Forms)
 - Browser Chrome version 14 ขึ้นไป

1.4 ความสำคัญของการวิจัย

1.4.1 ได้ผลเปรียบเทียบประสิทธิภาพด้านความเร็วของอัลกอริทึมบีบอัดแบบไม่สูญเสียข้อมูลที่เหมาะกับหมวดไฟล์เอกสาร หมวดไฟล์รูปภาพและหมวดไฟล์เพลงและวิดีโอ

1.4.2 ได้ต้นแบบการบีบอัดข้อมูลผ่านเว็บแอปพลิเคชัน โดยมีการบีบอัดข้อมูลอัตโนมัติตามผลการบีบอัดที่เหมาะสมของแต่ละหมวดหมู่เอกสารและช่วงขนาดไฟล์ข้อมูล

1.5 นิยามศัพท์เฉพาะ

1. CODEC คือ กลไกสำหรับเข้ารหัส และถอดรหัสข้อมูลดิจิทัลเมื่อเข้ารหัสด้วย CODEC ชนิดใด ก็จะต้องถอดรหัสด้วย CODEC ชนิดเดียวกัน
2. Dictionary คือ วิธีการเปลี่ยนข้อมูลที่มีความยาวคงที่ ไปเป็นรหัสที่มีความยาวคงที่แทน
3. Binary Tree คือ โครงสร้างข้อมูลแบบต้นไม้ เป็นวิธีการค้นหาข้อมูลเพื่อทำการบีบอัดข้อมูล
4. Run Length Encoding คือการสร้าง Dictionary เก็บข้อมูลที่มีค่าซ้ำ ๆ กันในแต่ละช่วงแถว



บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

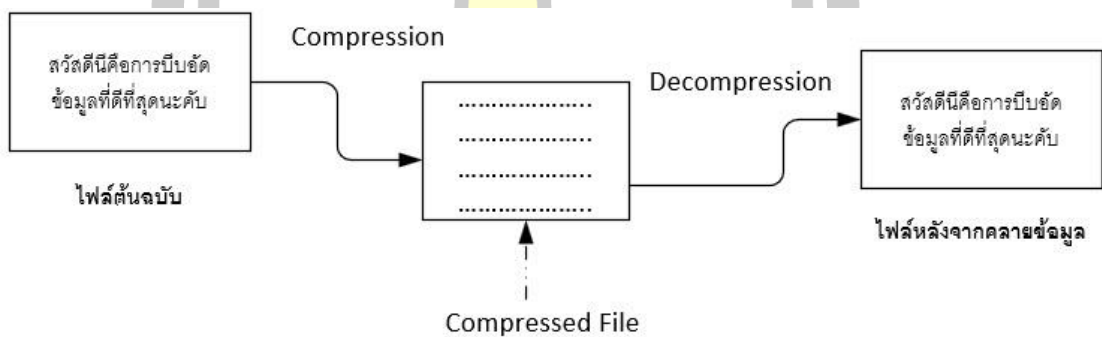
ในบทนี้จะกล่าวถึงทฤษฎีของการบีบอัดข้อมูล เพื่อลดขนาดข้อมูลของประเภทไฟล์ต่าง ๆ ซึ่งผู้วิจัยได้ศึกษาเกี่ยวกับทฤษฎีและงานวิจัยที่เกี่ยวข้องนำมาประยุกต์ใช้งานกับงานวิจัยนี้

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 การบีบอัดข้อมูล

การบีบอัดข้อมูล โดยทั่วไปประกอบด้วยการอ่านสายข้อมูลเข้ามาแล้วทำการแปล

ข้อมูลเหล่านี้ให้อยู่ในรูปแบบของรหัสต่าง ๆ ถ้าหากการบีบอัดข้อมูลมีประสิทธิภาพ รหัสข้อมูลที่ได้จะต้องมีขนาดที่สั้นกว่าข้อมูลเดิม



ภาพที่ 2.1 รูปแบบการบีบอัดข้อมูล

เทคนิคการบีบอัดเพื่อลดขนาดของข้อมูลแบ่งออกได้เป็น 2 ประเภท ดังนี้

2.1.1.1 เทคนิคและหลักการประมวลผลภาษาธรรมชาติ

เป็นเทคนิคในการศึกษาสถิติการใช้ตัวอักษร การใช้พยางค์ การใช้คู่ตัวอักษร เป็นต้น สามารถแบ่งออกเป็น 2 ชนิด ดังนี้

- แบบสถิต (Static) วิธีนี้จะต้องอ่านข้อมูล 2 ครั้ง ครั้งแรกเพื่อหาค่าความถี่ของตัวอักษรแต่ละตัวในข้อมูล และครั้งที่สองเพื่อแทนค่าตัวอักษรต่าง ๆ ด้วยรหัส วิธีการสร้างตารางจัดเก็บค่าความถี่จะคงที่และในการเข้ารหัสจะต้องเก็บตารางคำนวณความถี่เอาไว้ใช้ในการถอดรหัส เช่น วิธีฮัฟแมน วิธีนี้มีข้อเสียคือ ถ้าอักษรที่ปรากฏในข้อมูลไม่ซ้ำกันเลยจะทำให้ข้อมูลที่ผ่านการเข้ารหัสที่มีขนาดใหญ่กว่าข้อมูลก่อนการเข้ารหัส และตารางรหัสที่ใช้ก็เฉพาะเจาะจงสำหรับแต่ละแฟ้มข้อมูลเท่านั้น ข้อที่ควรคำนึงอีกประการ คือ การจัดเก็บตารางรหัสในลักษณะ order-0 ทำให้

สามารถเก็บรหัสได้เพียง 256 ไบต์ แต่ถ้าเปลี่ยนไปใช้ order – 1 จะสามารถเก็บรหัสได้ถึง 65, 536 ไบต์แต่ก็ทำให้เกิดโอเวอร์เฮดมาก

- แบบอะแดปทีฟ (Adaptive) วิธีนี้จะอ่านข้อมูลเข้ามาเพียงครั้งเดียวและตารางจัดเก็บค่าความถี่จะเปลี่ยนไปตามข้อมูลนำเข้าซึ่งจะใช้วิธีเดียวกันทั้งการเข้ารหัสและถอดรหัส จึงไม่จำเป็นต้องเก็บตารางรหัสไว้พร้อมกับข้อมูลเข้ารหัส เช่น วิธีอะแดปทีฟฮัฟแมน

2.1.1.2 หลักของพจนานุกรม

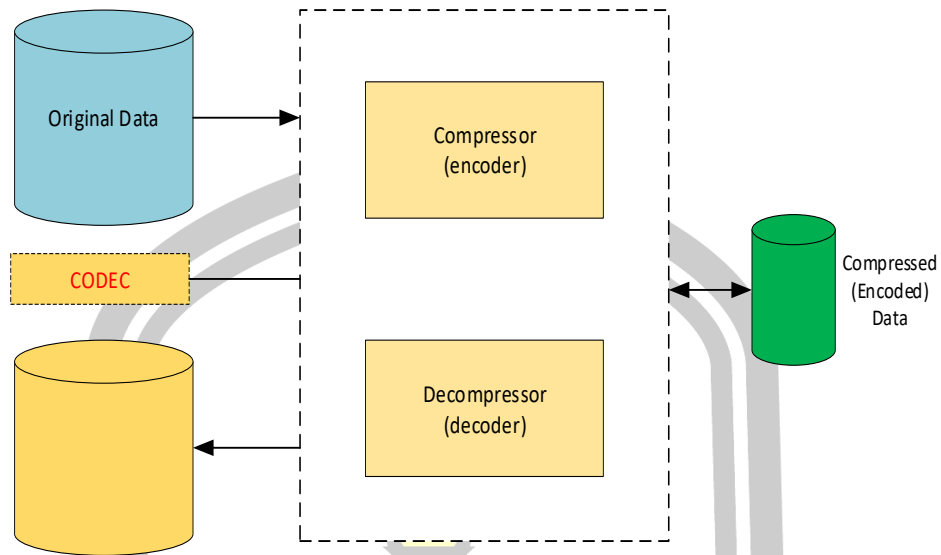
วิธีใช้หลักของพจนานุกรมนี้ต่างกับวิธีใช้ค่าทางสถิติคือ ในการเข้ารหัสของวิธีการสถิติจะเข้ารหัสทีละตัวอักษรแต่วิธีพจนานุกรมนี้จะอ่านข้อมูลจากแฟ้มข้อมูลเข้ามาและตรวจสอบกับข้อมูลที่อ่านเข้ามาก่อนหน้าถ้าหากพบอักษรหรือวลีที่ตรงกันจะให้ตัวชี้ที่ชี้ไปยังจุดที่ซ้ำกันนั้น ทำให้การบีบอัดข้อมูลทำให้ดีขึ้น วิธีใช้หลักของพจนานุกรม สามารถแบ่งออกได้เป็น 3 แบบ ดังนี้

- แบบสถิติ (Static) วิธีการนี้ใช้รูปแบบการบีบอัดข้อมูลและขยายข้อมูลคงที่ ไม่แปรตามข้อมูลที่เข้ารหัส ตารางเทียบรหัสหรือพจนานุกรมของคำหรือกลุ่มตัวอักษรคงที่ตอนเข้ารหัสและถอดรหัสวิธีนี้ข้อด้อย คือ ถ้าข้อมูลที่มีความแตกต่างกันมากตารางเทียบรหัสจะโตมากตามไปด้วย

- เซมิอะแดปทีฟ (Semiadaptive) วิธีนี้การขยายข้อมูลจะใช้ตารางเทียบรหัสซึ่งมีขนาดเล็กกว่า ข้อด้อยของวิธีนี้คือ ใช้เวลาในการอ่านและตรวจสอบข้อมูลว่าเป็นข้อมูลประเภทใดก่อนแต่ยังต้องมีการเก็บตารางเทียบรหัสไว้ในด้วย

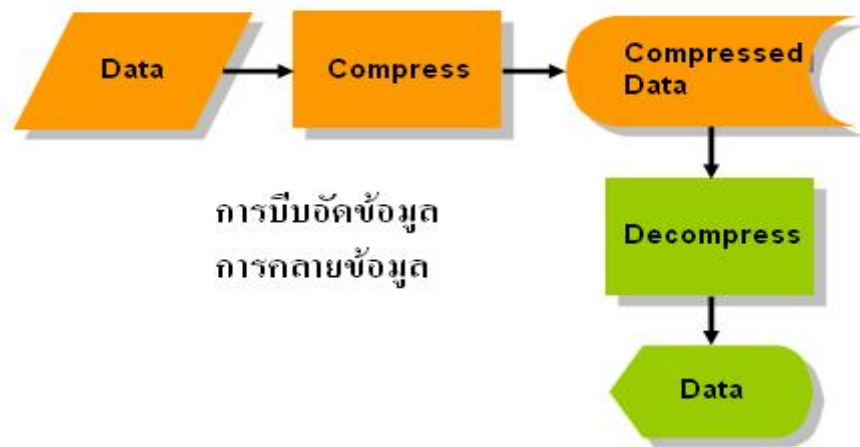
- อะแดปทีฟ (Adaptive) วิธีนี้ทั้งรูปแบบการเข้ารหัสและถอดรหัสจะใช้ตารางเทียบรหัสซึ่งได้จากข้อความที่ผ่านมาแล้ว เช่น วิธีแอลแซดดับบลิว

CODEC (Coder-Decoder) เป็นกลไกสำหรับเข้ารหัส และถอดรหัสข้อมูลดิจิทัลเมื่อเข้ารหัสด้วย CODEC ชนิดใด ก็จะต้องถอดรหัสด้วย CODEC ชนิดเดียวกัน CODEC จะใช้อัลกอริทึมในการหาข้อมูลที่มีความซ้ำซ้อนกัน (Redundant) และลดความซ้ำซ้อนนั้นลง โดยไม่ ส่งผลกระทบต่อคุณภาพของข้อมูล ข้อมูลที่ถูกบีบอัดแล้ว ก็จะมีนามสกุลของไฟล์แตกต่างกันไปตามวิธีการบีบอัด



ภาพที่ 2.2 ข้อมูลกลไกการรหัสด้วย CODEC

การบีบอัดข้อมูล (Data Compression) เป็นกระบวนการเข้ารหัสข้อมูลเพื่อให้ใช้จำนวนบิตในการเก็บข้อมูลน้อยกว่าเดิม การบีบอัดข้อมูลช่วยลดปริมาณการใช้ทรัพยากร เช่น ใช้แบนด์วิดท์ของระบบเครือข่ายน้อยลง เมื่อข้อมูลถูกบีบอัด (Compress) แล้วก็ต้องมีการคลาย (Decompress) หรือถอดรหัสเพื่อให้ได้ข้อมูลเดิมกลับมา



ภาพที่ 2.3 Flows การบีบอัดข้อมูล/คลายข้อมูล

ที่มา: <http://www.parinya.net/node/405>

การบีบอัดข้อมูล มี 2 รูปแบบ ดังนี้

1. การบีบอัดแบบสูญเสียข้อมูลไปบางส่วน (Lossy Compression) เป็นการบีบอัดข้อมูลโดยตัดข้อมูลต้นฉบับบางส่วนออกเพื่อลดขนาดไฟล์ โดยข้อมูลที่ซ้ำซ้อนจะถูกตัดทิ้งอย่างถาวร ทำให้ผลลัพธ์ที่ได้จากการบีบอัดไม่สมบูรณ์เหมือนกับข้อมูลต้นฉบับและคุณภาพของข้อมูลลดลงด้วย

2. การบีบอัดข้อมูลแบบที่ไม่มีการสูญเสียข้อมูล (Lossless Compression) เมื่อบีบอัดข้อมูลแล้วคลายข้อมูลกลับจะต้องได้ข้อมูลเหมือนเดิมครบถ้วนทุกประการด้วยเหตุนี้การบีบอัดข้อมูลแบบนี้จึงมักถูกใช้กับข้อมูลประเภทที่ข้อมูลจะสูญเสียไม่ได้ เช่น ข้อมูลประเภทข้อความหรือ แฟ้มข้อมูล โปรแกรม เป็นต้น การบีบอัดข้อมูลรูปภาพก็ใช้แนวทางการบีบอัดแบบนี้โดยอาศัย หลักการลดการซ้ำซ้อนของข้อมูล (Redundancy Reduction) และการใช้จำนวนบิตที่เก็บค่าข้อมูลไม่คงที่ (Variable-Length Coding) เช่น การเข้ารหัสแบบฮัฟแมน (Huffman Coding) และที่ใช้กันแพร่หลายก็คือการบีบอัดข้อมูลในรูปแบบแฟ้มข้อมูลรูปภาพของ PCX, GIF และ BMP ซึ่งให้อัตราการบีบอัดระหว่าง 10% ถึง 90 % สำหรับรูปภาพกราฟิก

2.1.2 เทคนิคฮัฟแมน โค้ด (Huffman Codes)

การเข้ารหัสแบบฮัฟแมนเป็นการใช้รหัสที่สั้นกว่าแทนสัญลักษณ์ที่เกิดขึ้นบ่อย โดยจะใช้ต้นไม้สองทางในการสร้างรหัสของสัญลักษณ์แต่ละตัวกระจายไปกับข้อมูลของโหนดใบที่เชื่อมอยู่กับต้นไม้สองทาง (Binary Tree) แต่ละโหนดจะมีน้ำหนักกำหนดอยู่ซึ่งก็คือความถี่หรือความน่าจะเป็นของการปรากฏของสัญลักษณ์นั้น ๆ [6] วิธีการสร้างต้นไม้สามารถทำได้ตามขั้นตอนดังนี้

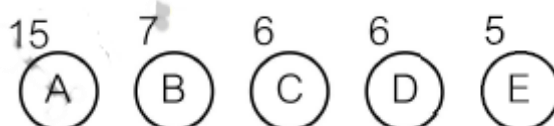
- 1) กำหนดให้ทุก ๆ สัญลักษณ์เป็นโหนดใด ๆ
- 2) หา 2 โหนดใด ๆ ที่มีน้ำหนักน้อยที่สุด
- 3) สร้างโหนดแม่สำหรับโหนดสองโหนดนี้โดยมีน้ำหนักเท่ากับผลรวมของน้ำหนัก ของโหนดลูก
- 4) กำหนดให้โหนดแม่นี้เป็นโหนดใด ๆ และนำโหนดลูกออกจากโหนดใด ๆ
- 5) โหนดลูกโหนดหนึ่งจะถูกกำหนดให้มีทางผ่านจากโหนดแม่เมื่อทำการถอดรหัสด้วยบิต 0 และอีกโหนดจะถูกกำหนดด้วย บิต 1
- 6) กลับไปข้อ 1) จนกว่าจะเหลือโหนดใด ๆ หนึ่งโหนดซึ่งโหนดนี้จะถูกกำหนดให้เป็นรากของต้นไม้

ตัวอย่างที่ 1 กำหนดให้มีข้อมูลต่าง ๆ ดังนี้

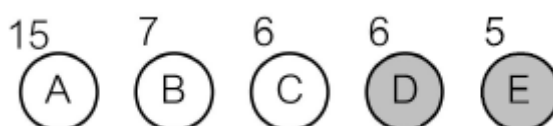
ตารางที่ 2.1 สัญลักษณ์และจำนวนของการปรากฏของข้อมูลสัญลักษณ์ ทั้งหมด 5 สัญลักษณ์คือ A, B, C, D และ E

Symbol	Count
A	15
B	7
C	6
D	6
E	5

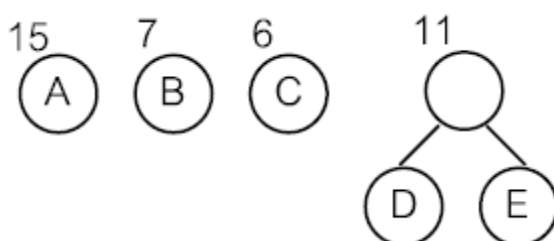
สัญลักษณ์ทั้ง 5 สัญลักษณ์นี้จะเป็นโหนดใบของต้นไม้ เมื่อกระบวนการเริ่มขึ้น โหนดทั้งหมดจะถูกกำหนดให้เป็นโหนดใด ๆ ในรายการ (List)



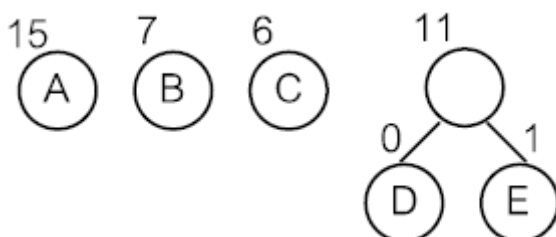
ขั้นแรก หา 2 โหนดที่มีน้ำหนักน้อยสุดคือ D และ E ซึ่งมีน้ำหนัก 6 และ 5



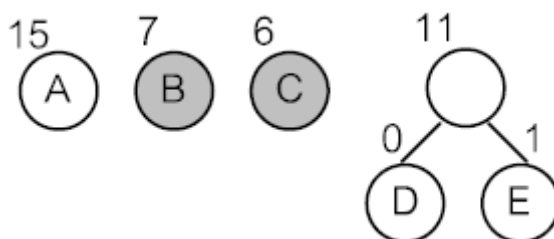
รวมโหนดทั้งสองเข้ากับโหนดแม่และกำหนดให้โหนดแม่มีน้ำหนักเป็น 11 และนำโหนด D และ E ออกจากโหนดใด ๆ ในรายการ



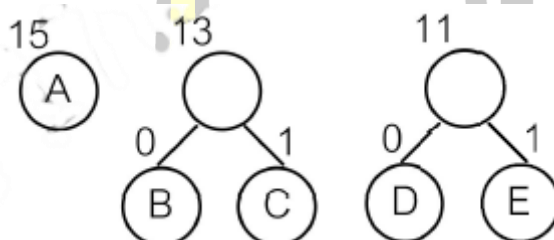
เมื่อขั้นตอนแรกเสร็จสิ้น เราจะดูว่าอะไรคือบิตที่น้อยที่สุดที่แสดงค่าของรหัส D และ E โดย D จะถูกแยกออกจากโหนดแม่ด้วย 0 (ซ้าย) และ E จะถูกแยกด้วย 1 (ขวา)



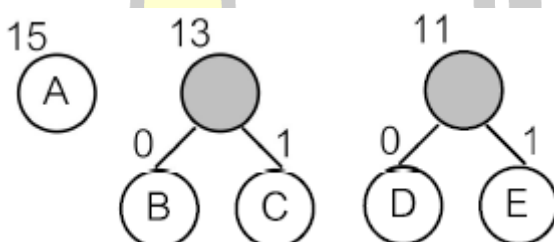
ขั้นต่อมา 2 โหนดที่มีค่าน้ำหนักน้อยที่สุดในโหนดใด ๆ คือโหนด B และ C



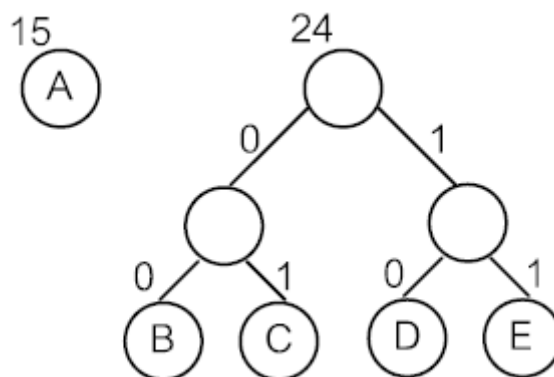
ดังนั้น ต้องเพิ่มโหนดแม่ให้ใหม่ และกำหนดให้มีน้ำหนักเป็น 13 และนำโหนด B และ C ออกจากโหนดใด ๆ ในรายการ



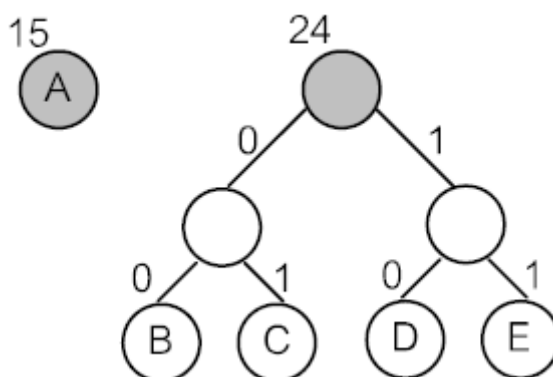
ขั้นต่อมา 2 โหนดที่มีค่าน้ำหนักน้อยที่สุดคือโหนดแม่ของ B/C และ D/E



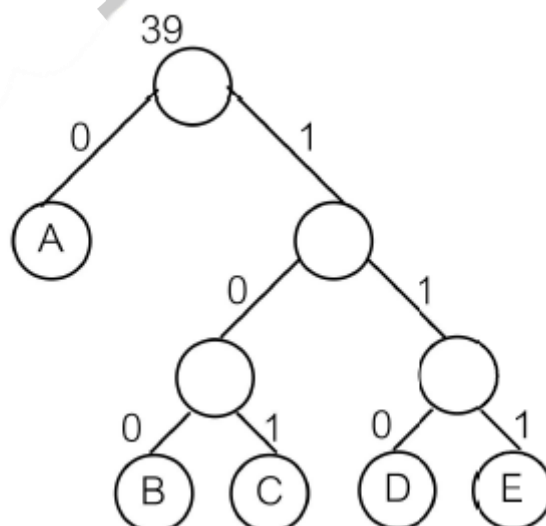
ดังนั้นต้องเพิ่มโหนดแม่ให้ใหม่ และกำหนดให้มีน้ำหนักเป็น 24 และนำโหนดลูกออกจากโหนดใด ๆ ในรายการ



ในขั้นตอนสุดท้ายจะเหลือเพียง 2 โหนดใด ๆ ในรายการ คือ โหนดแม่ที่มีน้ำหนัก 24 และโหนด A



สร้างโหนดแม่ให้ใหม่ให้มีน้ำหนักเป็น 39 และนำโหนดลูกออกจากโหนดใด ๆ ในรายการ
ตอนนี้เหลือโหนดแม่เพียงโหนดเดียว นั่นหมายถึงการสร้างต้นไม้เสร็จสิ้น



ตัวอย่างโครงสร้างโปรแกรมการเข้ารหัสแบบฮัฟแมน (Huffman Pseudocode)

พหุบัน ปณ กิจโต ชีเว

```

Huffman (C)
  n = the size of C
  insert all the elements of C into Q,
  using the value of the node as the priority
  for i in 1..n-1 do
    z = a new tree node
    x = Extract-Minimum (Q)
    y = Extract-Minimum (Q)
    left node of z = x
    right node of z = y
    f[z] = f[x] + f[y]
    Insert (Q, z)
  end for
  return Extract-Minimum (Q) as the complete tree

```

ภาพที่ 2.4 ตัวอย่างโครงสร้างโปรแกรมการเข้ารหัสแบบฮัฟแมน (Huffman Pseudocode)

ที่มา: <https://riptutorial.com/algorithm/example/23995/huffman-coding>

ในการกำหนดรหัสให้กับสัญลักษณ์นั้น ทำโดยเดินจากรากของต้นไม้ไปตามโหนดใบของต้นไม้ฮัฟแมน สะสมบิตให้ใหม่ถ้าเดินผ่านโหนดแม่แต่ละโหนดด้วยวิธีนี้จะได้โครงสร้างของรหัส แสดงได้ตามตารางที่ 2.2

ตารางที่ 2.2 แสดงรหัสแทนสัญลักษณ์ของการเข้ารหัสแบบฮัฟแมน

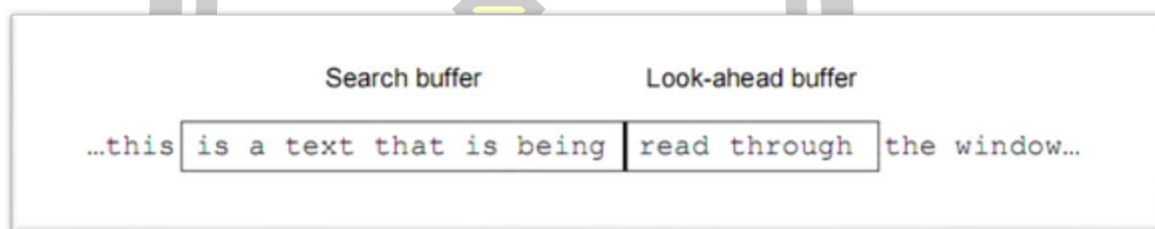
The Huffman Code Table	
A	0
B	100
C	101
D	110
E	111

จะเห็นว่ารหัสจะมีคุณสมบัติคือรหัสนำหน้าจะมีเพียงหนึ่งเดียว (Prefix code) ไม่ซ้ำกัน หมายความว่าไม่มีรหัสใดเป็นรหัสนำหน้าของรหัสตัวอื่น ๆ โดยรหัสฮัฟแมนสามารถถอดรหัสได้ อย่างไม่กำกวมเมื่อรหัสนั้นมาในรูปแบบสายอักขระ สัญลักษณ์ที่มีความถี่หรือค่าความน่าจะเป็นสูง ที่สุดคือ

A จะถูกกำหนดด้วยจำนวนบิตที่น้อยที่สุดและสัญลักษณ์ที่มีความถี่หรือค่าความน่าจะเป็นน้อยที่สุด คือ E จะถูกกำหนดด้วยจำนวนบิตที่มากที่สุด

2.1.3 เทคนิคแอลแซสเจ็ดเจ็ด (LZ77)

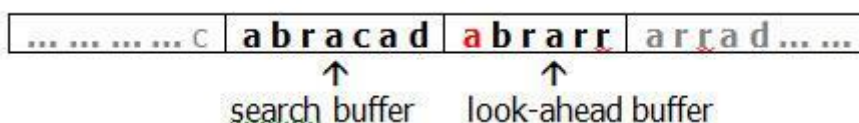
LZ77 เป็นวิธีบีบอัดที่มีการเก็บข้อมูลในอดีตที่ผ่านมาไม่นาน เพื่อนำไปใช้เป็น ข้อมูลอ้างอิงสำหรับการเปรียบเทียบข้อมูลที่กำลังเข้ารหัส คำรหัสที่ได้จากการเข้ารหัสเป็นตัวชี้ ตำแหน่งของข้อมูลในอดีตที่อยู่ก่อนหน้า การเข้ารหัส LZ77 ใช้ Sliding windows (ภาพที่ 2.5) ซึ่งเป็นหน้าต่างเสมือนที่เลื่อนไปตามแถวของข้อมูล ที่เวลาใดเวลาหนึ่งจะเห็นเฉพาะข้อมูลที่อยู่ตรงหน้า ต่างซึ่งประกอบด้วย บัฟเฟอร์ค้นหา (search buffer) และบัฟเฟอร์มองไปข้างหน้า (look ahead buffer)



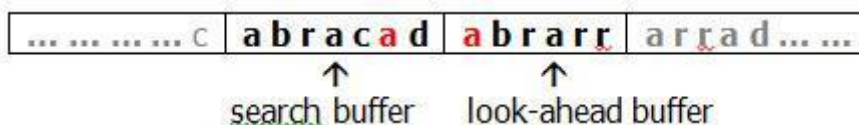
ภาพที่ 2.5 ตัวอย่างหน้าต่างของ sliding window

บัฟเฟอร์ค้นหาทำหน้าที่เป็น dictionary ส่วนบัฟเฟอร์มองไปข้างหน้า ทำหน้าที่เป็น keyword สำหรับค้นใน dictionary บัฟเฟอร์ทั้งสองเป็นส่วนหนึ่งของ sliding window จึงเลื่อนไปพร้อมกัน และขนาดของบัฟเฟอร์ทั้งสองนี้เป็นพารามิเตอร์ที่สำคัญในการประยุกต์ใช้งาน โดยมีสมมุติฐานว่า รูปแบบของข้อมูลที่ซ้ำกัน (pattern) เกิดขึ้นในพิสัยของบัฟเฟอร์ทั้งสองนี้ ผลของการเข้ารหัสจะแสดงในรูปแบบ $\langle \text{offset}, \text{length}, \text{character} \rangle$ โดย Offset คือ ระยะห่างนับจากตำแหน่งที่พบสัญลักษณ์ที่ตรงกันครั้งแรก Length คือ จำนวนของสัญลักษณ์ที่ต่อเนื่องกันในบัฟเฟอร์ที่เหมือนกัน และ Character คือ สัญลักษณ์ที่อยู่ในบัฟเฟอร์มองไปข้างหน้าซึ่งอยู่ ณ ตำแหน่งถัดมาจากกลุ่มสัญลักษณ์ที่ตรงกัน

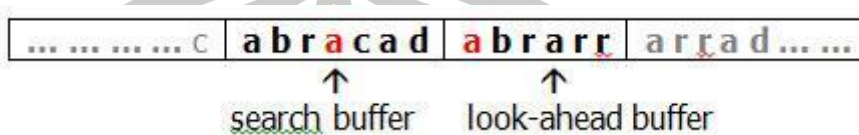
ตัวอย่างหลักการของ LZ77 (การบีบอัดข้อมูลตามขั้นตอนวิธีของ Jacob Ziv และ Abraham Lempel (LZ77), n.d.)



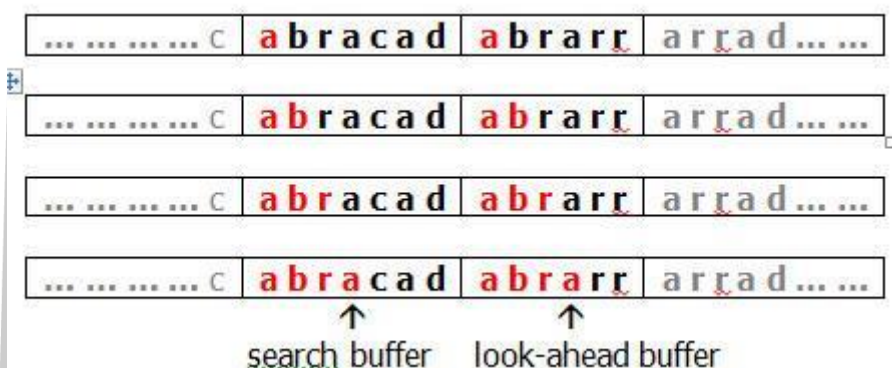
ครั้งแรกพบสายอักขระที่ตรงกันที่ $\text{offset} = 2, \text{length} = 1$



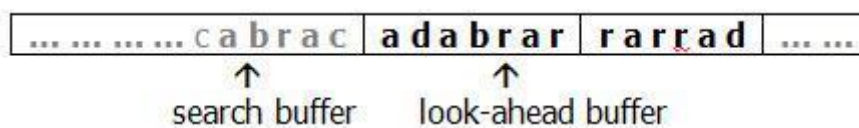
ครั้งที่สอง พบสายอักขระที่ตรงกันที่ offset = 4, length = 1, ความยาวเท่าครั้งแรก



ครั้งที่สาม พบสายอักขระที่ตรงกันที่ offset = 7, length = 4, ความยาวสูงกว่า จึงเปลี่ยนแทนคู่ลำดับ (offset, length) ที่เก็บไว้เดิมด้วยค่าใหม่

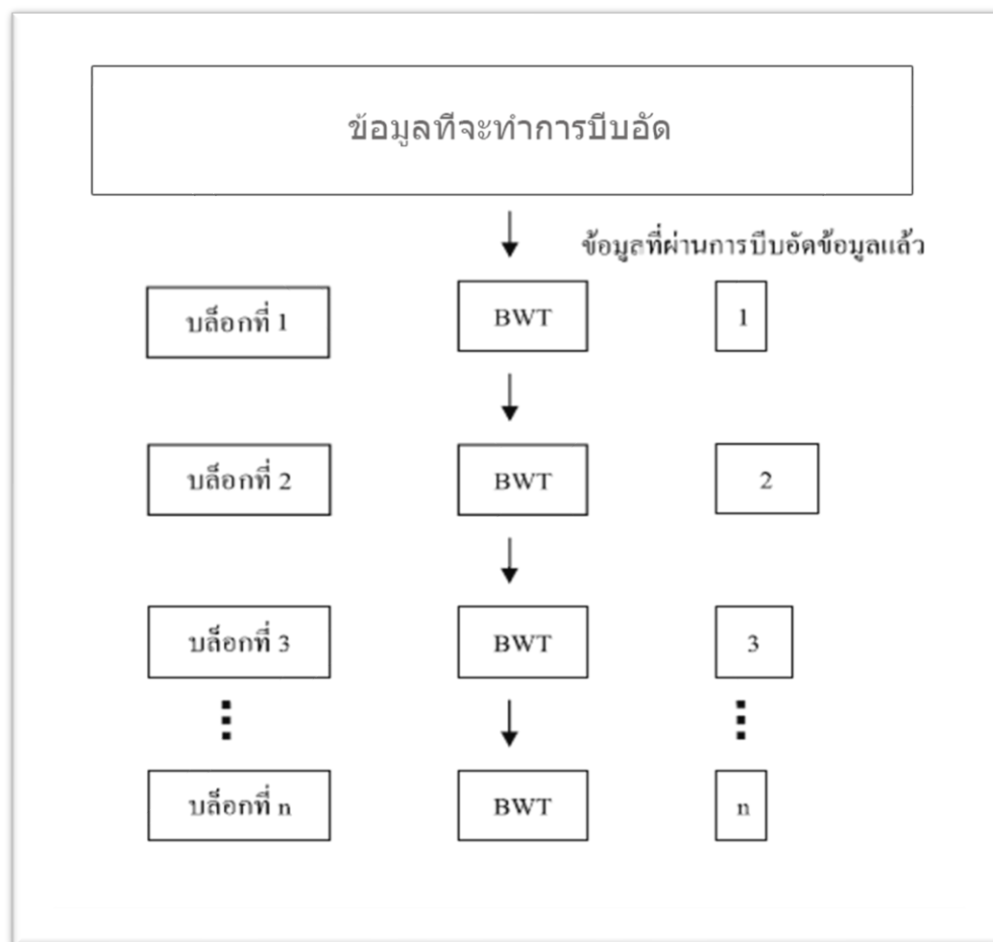


ตัวเข้ารหัสส่งข้อมูลออกเป็น (7, 4, r) จากนั้นจึงเลื่อน sliding window ไปทางขวา length + 1 = 5



2.1.4 เทคนิคบีบอัด (BZip2)

บีบอัดเป็นโปรแกรมบีบอัดข้อมูลโอเพ่นซอร์ส (Open Source) ที่มีความนิยมและหาใช้ได้ง่าย ในระบบยูนิกซ์ใช้วิธีการเบอร์โรวส์-วีเลอร์และการเข้ารหัสแบบฮัฟแมน โดยปกติการบีบอัดข้อมูลแบบนี้จะดีกว่าการบีบอัดข้อมูลแบบ LZ77/LZ78 และมีประสิทธิภาพใกล้เคียงแบบ PPM ของการบีบอัดข้อมูลโดยใช้สถิติแต่เร็วกว่าทางด้านเวลาโปรแกรมจะประมวลผลข้อมูลโดย แบ่งข้อมูลออกเป็นช่วงบล็อกขนาด 100,000 ไบต์ถึง 900,000 ไบต์ขึ้นอยู่กับคำสั่ง ปกติขนาดของ บล็อกจะเท่ากับ 900,000 ไบต์โปรแกรมจะอ่านข้อมูลครั้งละ 5,000 ไบต์จนกว่าจะครบเท่ากับขนาดของบล็อกที่กำหนดและประมวลผลบีบอัดข้อมูลเขียนข้อมูลที่บีบอัดแล้วลงบนหน่วยเก็บข้อมูล ทำเช่นนี้ต่อไปเรื่อย ๆ จนกว่าบล็อกข้อมูลถูกบีบอัดข้อมูลแล้วทั้งหมดดังแสดงในภาพที่ 2.6

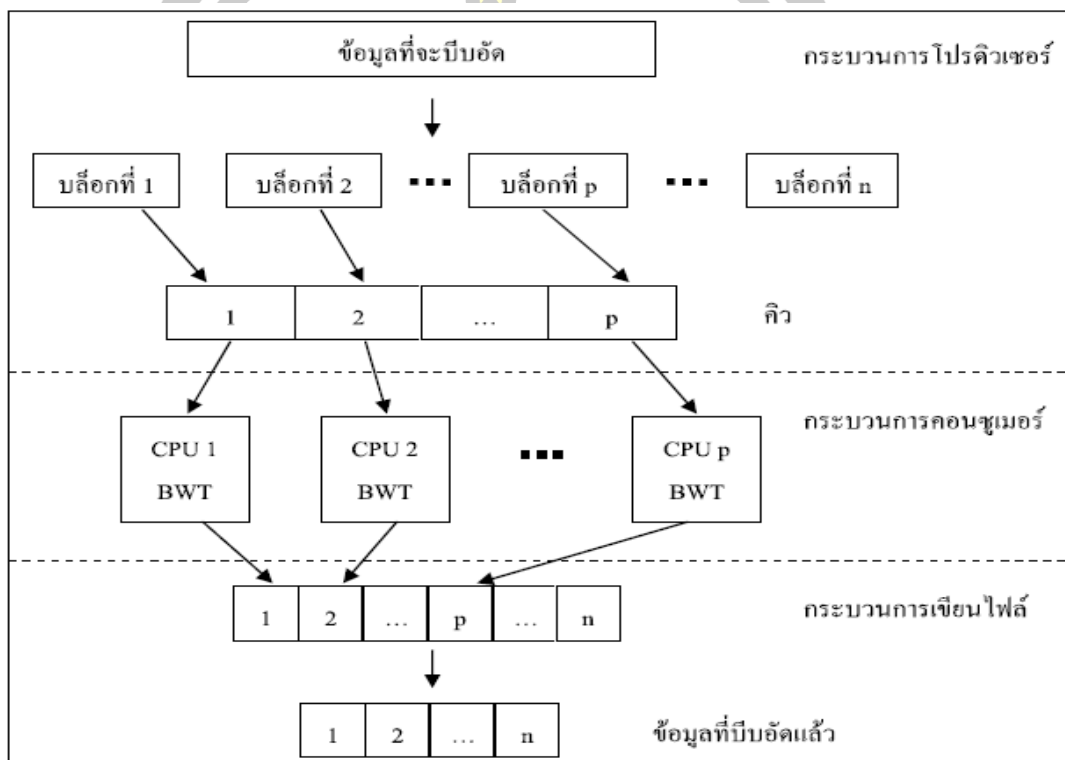


ภาพที่ 2.6 แสดงการบีบอัดข้อมูลด้วยวิธีบีบอัด

จากภาพที่ 2.6 จะพบว่า การบีบอัดข้อมูลด้วยวิธีบีบอัดข้อมูลนี้เหมาะสมกับการนำไปพัฒนาเป็นการบีบอัดข้อมูลแบบขนาน เนื่องจากข้อมูลที่จะทำการบีบอัดข้อมูลถูกแบ่งออกเป็นบล็อกข้อมูลย่อยหลาย ๆ บล็อก จากนั้นบล็อกข้อมูลย่อยแต่ละบล็อกจะผ่านกระบวนการบีบอัดข้อมูล ซึ่งกระบวนการบีบอัดข้อมูลนี้จะถูกใช้เหมือนกันกับทุก ๆ บล็อกข้อมูลย่อย ดังนั้น บล็อกข้อมูลย่อยแต่ละบล็อก สามารถนำมาประมวลผลบีบอัดข้อมูลพร้อมกันได้ในระบบประมวลผลแบบขนาน

การบีบอัดข้อมูลแบบขนานด้วยวิธีบีบอัดข้อมูลนี้ถูกพัฒนาขึ้นในปี ค.ศ. 2004 โดย เจฟฟ์ กิลคริสต์ (Jeff Gilchrist) ซึ่งพัฒนาการบีบอัดข้อมูลนี้บนเครื่องคอมพิวเตอร์แบบขนานที่ใช้หน่วยความจำรวม (Shared-Memory Parallel Architectures) วิธีนี้จะแบ่งข้อมูลออกเป็นบล็อกเท่า ๆ กัน ซึ่งกำหนดโดยผู้ใช้แทนที่จะอ่านข้อมูลครั้งละ 5,000 ไบต์มีการนำระบบคิวและแบบจำลอง โพรดิวเซอร์-คอนซูเมอร์ (Producer-Consumer Model) ใช้ในการออกแบบพัฒนาระบบ โดยการประมวลผลแบบขนานจะประกอบด้วยคอนซูเมอร์หลาย ๆ คอนซูเมอร์ขึ้นอยู่กับหน่วย

ประมวลผล โดยโปรติวเซอร์จะทำหน้าที่แบ่งข้อมูลออกเป็นบล็อกแล้วนำเข้าคิวและคอนซูเมอร์ จะทำหน้าที่ อ่านข้อมูลจากคิว และประมวลผลบีบอัดข้อมูลด้วยวิธีบีบอัดจากนั้นข้อมูลที่ถูกลบีอัดแล้วจะถูก เขียนลงไฟล์ผลลัพธ์ให้ถูกต้องตามลำดับโดยกระบวนการเขียนไฟล์ดังแสดงในรูป 2.7 เมื่อแต่ละคอนซูเมอร์ทำแต่ละกระบวนการเสร็จแล้วจะไปอ่านข้อมูลจากคิว ทำเช่นนี้ไปเรื่อย ๆ จนกว่าคิวจะว่าง นั่นคือข้อมูลทั้งหมดถูกลบีอัดข้อมูลแล้ว



ภาพที่ 2.7 แสดงการบีบอัดข้อมูลแบบขนานด้วยวิธีบีบอัด

2.1.5 เทคนิคดีเฟต (DEFLATE)

Deflate เป็นอัลกอริทึมการบีบอัดข้อมูลแบบไม่สูญเสียและรูปแบบไฟล์ที่เชื่อมโยงซึ่งใช้การรวมกันของอัลกอริทึม LZ77 และการเข้ารหัส Huffman อัลกอริทึม Deflate ถูกกำหนดโดย Phil Katz สำหรับเป็นเครื่องมือจัดเก็บไฟล์แบบบีบอัดข้อมูล (Archiving Tool) PKZIP เวอร์ชันรูปแบบไฟล์ถูกระบุใน RFC 1951 อัลกอริทึมดั้งเดิมที่ออกแบบโดย Katz ได้รับการจดสิทธิบัตรเป็นสิทธิบัตรสหรัฐอเมริกา 5,051,745 และมอบให้ PKWARE, Inc. ตั้งที่ระบุไว้ในเอกสาร RFC อัลกอริทึมที่สร้างไฟล์ Deflate นั้นถูกนำมาใช้ในลักษณะที่ไม่ครอบคลุมโดยสิทธิบัตร สิ่งนี้นำไปสู่การใช้กันอย่างแพร่หลายตัวอย่างเช่นในไฟล์บีบอัด GZIP, ไฟล์ภาพ PNG และรูปแบบไฟล์ ZIP

DEFLATE จะตัดสตริงออกเป็นบล็อก บล็อกละ 32kB โดยแต่ละบล็อกนั้นจะใช้พื้นที่ 4 ไบต์ในการเก็บระยะทาง (Distance Code) โดยเป็นระยะทางที่จะถูกแทนที่และสามารถเก็บ Symbol ที่จะถูกแทนที่ 288 Symbols หลักการทำงานคือ จะตัดข้อมูลออกเป็นบล็อกย่อย ๆ บล็อกละ 32kB ก่อน หลังจากนั้นแต่ละบล็อกจะใช้อัลกอริทึม LZ77 ในการหาบล็อกย่อย ๆ ที่มีบิตซ้ำกันในลักษณะที่เรียกว่า Sliding Window หลังจากนั้นจะสร้างตารางเพื่อเก็บ Symbol ขึ้นมา ซึ่งลำดับของ Symbol แสดงในตารางที่ 2.3

ตารางที่ 2.3 การใช้งานของตาราง Symbol ในอัลกอริทึม DEFLATE

Symbol ตัวที่	การใช้งาน
0-255	Symbol 256 ตัวที่มีการใช้งานทั่วไป
256	เพื่อแสดงว่าจบ Data Stream ให้หยุดการทำงาน
257-285	เป็นการรวมบิตพิเศษ ประกอบด้วยความยาว 3-258 ไบต์
286 และ 287	ไม่มีการใช้งาน (Reserved)

หลังจากนั้นจะสร้างตาราง Distance Code เพื่อเก็บระยะทางที่จะเข้าไปแทนที่ Symbol ที่ระบุไว้ในตัวข้อมูล อัลกอริทึม DEFLATE นี้ได้ผลดีมากในการบีบอัดข้อมูลที่เป็นลักษณะแบบนารี เช่น ซอฟต์แวร์ต่าง ๆ จนนำไปพัฒนาต่อมีการใช้งานที่หลากหลายเช่น LZH ซึ่งใช้ในการบีบอัดภาพประเภท GIF แต่วิธีนี้ไม่เหมาะกับการบีบอัดไฟล์ขนาดเล็กและเท็กซ์ไฟล์เนื่องจากทำงานได้ช้ากว่าวิธีแบบ Entropy Encoding เช่น Huffman Coding และต้องเสียพื้นที่ในการสร้าง Header ต่าง ๆ ในแต่ละบล็อกมากกว่าวิธีอื่น ๆ

2.1.6 เทคนิคแอลแซสโฟร์ (LZ4)

LZ4 (Collet, 2015) เป็นอัลกอริทึมที่อยู่ในตระกูล LZ77 ใช้ในการบีบอัดข้อมูลซึ่งมุ่งเน้นที่ความเร็วของการบีบอัดข้อมูลและถอดรหัสข้อมูล ค้นหาที่ซ้ำซ้อนกัน โดยอาศัย Hash table และไม่ค้นหาความเป็นไปได้ทั้งหมดของค่าที่ซ้ำซ้อนกัน และทำการบีบอัดกับข้อมูลในระดับไบต์

LZ4 มีความเร็วในการบีบอัดข้อมูลอยู่ที่ 400 MB/s ต่อคอร์ เหมาะกับการใช้งานของซีพียูที่มีหลายคอร์ และสามารถแปลงข้อมูลกลับคืนมาด้วยความเร็วหลาย GB/s ต่อคอร์ ซึ่งโดยปกติแล้วความเร็วในการแปลงข้อมูลกลับคืนนั้นสามารถเร็วได้ถึงความเร็วของแรมซึ่งถูกจำกัดบนระบบการทำงานหลายคอร์

ตารางที่ 2.4 ผลการวัดประสิทธิภาพของ LZ4 เมื่อเทียบกับเครื่องมือในการบีบอัดอื่น ๆ

Compressor	Ratio	Compression	Decompression
memcpy	1.000	4200 MB/s	4200 MB/s
LZ4 fast 17 (r129)	1.607	690 MB/s	2220 MB/s
LZ4 default (r129)	2.101	385 MB/s	1850 MB/s
LZO 2.06	2.108	350 MB/s	510 MB/s
QuickLZ 1.5.1.b6	2.238	320 MB/s	380 MB/s
Snappy 1.1.0	2.091	250 MB/s	960 MB/s
LZF v3.6	2.073	175 MB/s	500 MB/s
zlib 1.2.8 -1	2.730	59 MB/s	250 MB/s
LZ4 HC (r129)	2.720	22 MB/s	1830 MB/s
zlib 1.2.8 -6	3.099	18 MB/s	270 MB/s

ที่มา: [https://en.wikipedia.org/wiki/LZ4_\(compression_algorithm\)](https://en.wikipedia.org/wiki/LZ4_(compression_algorithm))

LZ4 Sequence

Token ::= 4-high-bits : literal length / 4-low-bits : match length

Token	Literal length+ (optional)	Literals	Offset	Match length+ (optional)
1-byte	0-n bytes	0-L bytes	2-bytes (little endian)	0-n bytes

ภาพที่ 2.8 ส่วนประกอบของบล็อกการบีบอัดข้อมูล (Compression block) ของ LZ4

ที่มา: [https://en.wikipedia.org/wiki/LZ4_\(compression_algorithm\)](https://en.wikipedia.org/wiki/LZ4_(compression_algorithm))

ข้อมูลที่ถูกรบีบอัด (Compression block) ของ LZ4 นั้นประกอบด้วย sequence หลาย ๆ sequence เรียงต่อกัน ในแต่ละ sequence ดังแสดงในภาพที่ 2.8 นั้น คือชุดของ literals ซึ่งเป็นส่วนที่ไม่ถูกรบีบอัดหรือข้อมูลที่ไม่ซ้ำ ตามด้วยส่วนที่ใช้ในการก๊อปปี้ match (ส่วนที่ระบุถึงการซ้ำของข้อมูล) ในแต่ละ sequence จะเริ่มต้นด้วย token ซึ่งมีขนาด 1 ไบต์ ซึ่งใน 1 ไบต์นี้จะแยกออกมาเป็น 2 ฟิลด์ ซึ่งแต่ละฟิลด์มีขนาด 4 บิต และมีค่าอยู่ระหว่าง 0-15 ค่าใน token นั้นจะใช้สำหรับระบุความยาวและส่วนประกอบต่าง ๆ ใน sequence นั้น โดย 4 บิตแรก (high-bits) นั้นจะบอกถึงความ

ยาวของ literal และ 4 บิตหลัง (low-bits) นั้นจะบอกถึงความยาวของ match ซึ่งความยาวของ literal และ match นั้น ถ้าหากว่าเกินกว่า 15 จะมีส่วน literal length เพิ่มขึ้นมา เพื่อเก็บค่าความยาวของ literal โดยแต่ละไบต์จะเก็บค่าได้ 255 เมื่อ literal length มีความยาวมากขึ้น ก็สามารถเพิ่มขนาดของ literal length ได้ โดยไม่มีจำกัดขนาดไว้ (no size limit) จาก literals คือ operation ที่ใช้ในการก๊อปปี้ match ซึ่งเริ่มจาก offset ซึ่งมี ขนาด 2 ไบต์ แบบ little endian (ไบต์แรกคือ low ไบต์ที่ 2 คือ high) ระบุถึงตำแหน่งของ match ที่ถูกก๊อปปี้มาจาก literals ซึ่ง 1 หมายถึงตำแหน่งปัจจุบัน ลบ 1 byte และค่าสูงสุดของ offset คือ 65535

ในการคำนวณความยาวของ match จะใช้ฟิลด์ที่สองของ token ซึ่งเป็น 4 บิตหลัง (low-bits) โดยความยาวที่น้อยที่สุดของ match เท่ากับ 4 ไบต์ เรียกว่า min match ดังนั้น ค่า 0 จึงหมายถึง 4 ไบต์ และ 15 หมายถึง 19+ ไบต์ ซึ่งสามารถขยายให้มากขึ้นได้เช่นเดียวกับ literal decoder สามารถสร้างข้อมูลต้นฉบับด้วยข้อมูลจาก offset และความยาวของ match

2.1.7 เทคนิคแอลแซสเอ็มเอ (LZMA)

เป็นเทคนิคการบีบอัดข้อมูลแบบ lossless มันถูกพัฒนาตั้งแต่ปี 1996 หรือ 1998 โดย Igor Pavlov และถูกใช้ครั้งแรกในรูปแบบ 7z ของ Archiver 7-Zip อัลกอริทึมนี้ใช้รูปแบบการบีบอัดแบบพจนานุกรมคล้ายกับอัลกอริทึม LZ77 ที่เผยแพร่โดย Abraham Lempel และ Jacob Ziv ในปี 1977 และมีอัตราส่วนการอัดสูง (โดยทั่วไปสูงกว่า bzip2) และขนาดพจนานุกรมการบีบอัดตัวแปร (สูงสุด 4 GB) ในขณะที่ยังคงรักษาความเร็วในการคลายการบีบอัดคล้ายกับอัลกอริทึมการบีบอัดอื่นๆ ที่ใช้กันทั่วไป

LZMA ใช้อัลกอริทึมการบีบอัดพจนานุกรม (ตัวแปรของ LZ77 ที่มีขนาดพจนานุกรมขนาดใหญ่และการสนับสนุนพิเศษสำหรับระยะการจับคู่ที่ใช้ซ้ำ ๆ) ซึ่งผลลัพธ์จะถูกเข้ารหัสด้วยตัวเข้ารหัสแบบเป็นช่วง โดยใช้แบบจำลองที่ซับซ้อนเพื่อทำนายความน่าจะเป็นของแต่ละบิต ตัวบีบอัดพจนานุกรมพบการจับคู่โดยใช้โครงสร้างข้อมูลพจนานุกรมที่ซับซ้อน และสร้างสัญลักษณ์ การอ้างอิงวลีซึ่งถูกเข้ารหัส ทีละหนึ่งบิต

ในการบีบอัด LZMA สตริงที่ถูกบีบอัดเป็นสตริงของบิตเข้ารหัสโดยใช้ Coder Range แบบปรับตัวได้ สตริงจะถูกแบ่งออกเป็นแพ็คเก็ตแต่ละแพ็คเก็ตที่อธิบายไบต์เดียวหรือลำดับ LZ77 ที่มีความยาวและระยะทางโดยนับหรือเข้ารหัสอย่างชัดเจน แต่ละส่วนของแต่ละแพ็คเก็ตถูกจำลองด้วยบริบทอิสระดังนั้นการทำนายความน่าจะเป็นสำหรับแต่ละบิตนั้นมีความสัมพันธ์กับค่าของบิตนั้น (และบิตที่เกี่ยวข้องจากเขตข้อมูลเดียวกัน) ในแพ็คเก็ตก่อนหน้าของประเภทเดียวกัน แพ็คเก็ตมี 7 ประเภทตามตารางที่ 2.5

ตารางที่ 2.5 คำอธิบายแพ็คเกจการจับเก็บข้อมูลอัลกอริทึม LZMA

รหัสที่บรรจุ (ลำดับบิต)	ชื่อแพ็คเกจ	คำอธิบายแพ็คเกจ
0 + byteCode	LIT	ไบต์เดียวเข้ารหัสโดยใช้ coder ช่วงแบบปรับตัวได้
1 + 0 + len + dist	การจับคู่	ลำดับ LZ77 ทั่วไปที่อธิบายความยาวและระยะทางของลำดับ
1 + 1 + 0 + 0	SHORTREP	ลำดับ LZ77 หนึ่งไบต์ ระยะทางเท่ากับระยะทาง LZ77 ที่ใช้ล่าสุด
1 + 1 + 0 + 1 + len	LONGREP [0]	ลำดับ LZ77 ระยะทางเท่ากับระยะทาง LZ77 ที่ใช้ล่าสุด
1 + 1 + 1 + 0 + len	LONGREP [1]	ลำดับ LZ77 ระยะทางเท่ากับระยะทาง LZ77 ที่ใช้ล่าสุดครั้งที่สอง
1 + 1 + 1 + 1 + 0 + len	LONGREP [2]	ลำดับ LZ77 ระยะทางเท่ากับระยะทาง LZ77 ที่ใช้ล่าสุดครั้งที่สาม
1 + 1 + 1 + 1 + 1 + len	LONGREP [3]	ลำดับ LZ77 ระยะทางเท่ากับระยะทาง LZ77 ที่ใช้ล่าสุด

ที่มา: https://en.wikipedia.org/wiki/Lempel-Ziv-Markov_chain_algorithm

ความยาวมีการเข้ารหัสตามตารางที่ 2.6

ตารางที่ 2.6 รายละเอียดรหัสความยาวข้อมูล

รหัสความยาว (ลำดับบิต)	ลักษณะ
0+ 3 บิต	ความยาวที่เข้ารหัสโดยใช้ 3 บิตจะให้ช่วงความยาวตั้งแต่ 2 ถึง 9
1 + 0 + 3 บิต	ความยาวที่เข้ารหัสโดยใช้ 3 บิตทำให้ความยาวอยู่ในช่วงตั้งแต่ 10 ถึง 17
1 + 1 + 8 บิต	ความยาวที่เข้ารหัสโดยใช้ 8 บิตให้ช่วงความยาวตั้งแต่ 18 ถึง 273

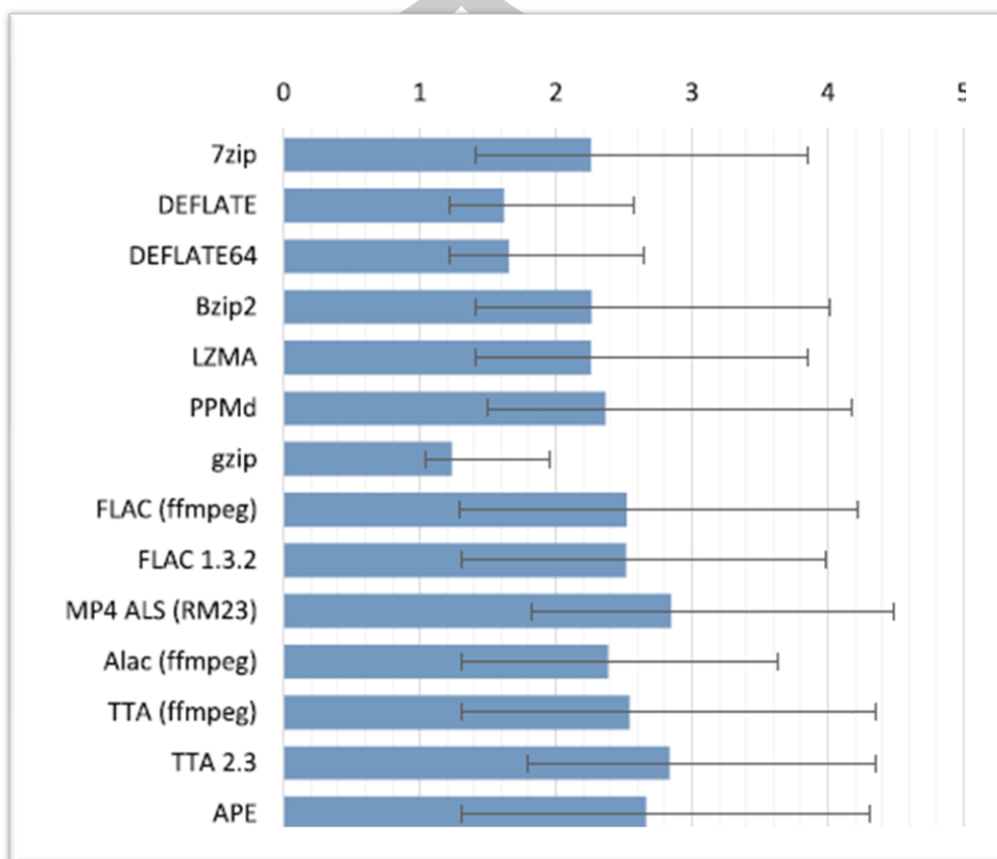
ที่มา: https://en.wikipedia.org/wiki/Lempel-Ziv-Markov_chain_algorithm

เช่นเดียวกับใน LZ77 ความยาวไม่ จำกัด โดยระยะทางเนื่องจากการตัดออกจากพจนานุกรมถูกกำหนดไว้ว่าการตัดลอกดำเนินการเป็นไบต์ต่อไบต์ทำให้ระยะทางคงที่ ระยะทางเป็น 32- บิตและระยะทาง 0 จุดถึงไบต์ที่เพิ่มล่าสุดในพจนานุกรม

2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 Jumar และคณะ [7] ได้ทำการเปรียบเทียบการบีบอัดแบบไม่สูญเสียข้อมูล เพื่อทำการวัดอัตราส่วนการบีบอัดและค้นหากลยุทธ์การบีบอัดที่เหมาะสมที่สุด โดยใช้การบีบอัดข้อมูล 2 หมวดหมู่คือ 1.อัลกอริทึมบีบอัดด้านเสียงโดยเฉพาะซึ่งประกอบไปด้วยอัลกอริทึม ALAC, ALS, APE, FLAC และ TrueAudio 2.อัลกอริทึมการบีบอัดข้อมูลทั่วไป ได้แก่ LZMA, Delfate, PPMd, BZip2 และ Gzip เพื่อนำผลการเปรียบเทียบมาใช้ในงานด้านระบบบริหารการใช้ไฟฟ้าอย่างมีประสิทธิภาพด้วยเทคโนโลยีด้านการสื่อสาร (Smart Power Grid) จากผลการวิจัยพบว่าอัลกอริทึม LZMA, 7Zip

และ Bzip2 มีผลลัพธ์ที่ดีกว่าทั้งวิธี DEFLATE และ Gzip ในหมวดของการบีบอัดทั่วไป แต่หมวดอัลกอริทึมด้านเสียงนั้น มีค่าความแตกต่างกันแค่นิดเดียว



ภาพที่ 2.9 ผลการเปรียบเทียบการบีบอัดข้อมูลทั้ง 2 หมวดหมู่

2.2.2 Ibrahim และ Gbolagade [8] ได้ทำการศึกษาการบีบอัดแบบไม่สูญเสียข้อมูล เพื่อทำการวัดขนาดการจัดเก็บไฟล์หลังจากมีการบีบอัดข้อมูล และความเร็วในการส่งข้อมูลและคุณภาพของข้อมูล ในงานวิจัยนี้ได้ใช้ข้อมูลรูปภาพจากฐานข้อมูลจำนวน 10 รูปภาพในการทดสอบ จากผลการทดสอบปรากฏว่า การบีบอัด CRT-LZW ช่วยประหยัดพื้นที่และการบีบอัดอย่างรวดเร็ว (หรือกำจัดความซ้ำซ้อน) ของภาพต้นฉบับมากกว่าการเข้ารหัส CRT-Huffman 29.78% ถึง 14.00% และเวลาในการบีบอัดวิธีการ CRT-LZW ทำได้ดีกว่าวิธี CRT-Huffman ประมาณ 9.95 วินาที ถึง 19.15 วินาที สำหรับอัตราส่วนการบีบอัด CRT-LZW ยังมีประสิทธิภาพสูงกว่าการเข้ารหัส CRT-Huffman โดย 0.39 db. ถึง 4.38 db.

2.2.3 ภควัฒน์ ดิณศิริสุข [2] งานวิจัยนี้ได้พัฒนาวิธีการบีบอัดข้อมูลเพื่อให้เหมาะสมกับพฤติกรรมการทำงานและลักษณะของข้อมูลบนระบบเครือข่ายเช่นเซอวีร์สาย โดยมีวัตถุประสงค์เพื่อ

ลดปริมาณการส่งข้อมูล ด้วยการดัดแปลงอัลกอริทึมที่มีอยู่แล้วให้สามารถทำงานได้ดีบนเครือข่าย เซนเซอร์ไร้สายซึ่งใช้ Huffman Algorithm, Adaptive Huffman Algorithm เป็นต้นแบบในการดัดแปลงอัลกอริทึม

2.2.4 Uthayakuma และคณะ [3] ได้นำวิธีการบีบอัดข้อมูลมาใช้ในเรื่อง wireless sensor networks (WSN) เนื่องจากโหนดเซนเซอร์ใช้พลังงานจากแบตเตอรี่ที่มีความจุจำกัด เนื่องจากการส่งข้อมูลเป็นกระบวนการที่ต้องใช้พลังงานหลักใน WSN จึงมีการนำเทคนิคการประหยัดพลังงานหลายรูปแบบมาใช้ และรูปแบบการบีบอัดข้อมูลเป็นเทคนิคที่ใช้พลังงานอย่างมีประสิทธิภาพซึ่งช่วยลดปริมาณข้อมูลที่จะส่งในเครือข่ายทำให้ประหยัดพลังงานได้มาก จากการทดลองการนำวิธีการบีบอัดข้อมูลมาใช้สามารถประหยัดพลังงานได้มากถึง 87.57%

2.2.5 Kasmeera และคณะ [9] ได้เลือกใช้อัลกอริทึม Huffman Codes พัฒนาโดย David Huffman ในปี ค.ศ. 1951 มาใช้ในการเข้ารหัสและบีบอัดข้อมูลรูปภาพเพื่อให้มีความปลอดภัย จากผลการทดลองแสดงให้เห็นว่าประสิทธิภาพการบิดเบือนอัตราส่วนการบีบอัดของวิธีนี้ดีกว่าเทคนิคที่มีอยู่ อัตราส่วนการบีบอัดของรูปภาพที่เข้ารหัสได้รับการปรับปรุงให้อยู่ในช่วงระหว่าง 10 ถึง 20 เปอร์เซ็นต์

2.2.6 Li, Wenwen และคณะ [10] ได้เลือกใช้อัลกอริทึม LZMA และ DEFLATE เพื่อมาใช้ในการบีบอัดข้อมูล, การคลายการบีบอัดข้อมูล ในระบบการจัดการภัยพิบัติ ผ่านเว็บเซอร์วิส เพื่อจัดการปัญหาด้านประสิทธิภาพขั้นพื้นฐานที่ส่งผลกระทบต่อภารกิจข้อมูลระยะไกล

2.2.7 Uthayakumar และ Vengattaraman [5] งานวิจัยนี้เป็นการเปรียบเทียบอัลกอริทึมเพื่อเปรียบเทียบการบีบอัดรูปภาพจำนวน 71 รูปภาพ และแต่ละภาพมีความบิดเบือนจากความเป็นจริงสูง และทำการเลือกอัลกอริทึมมาทดสอบ จำนวน 6 อัลกอริทึมดังนี้ Lempel Ziv Markov chain Algorithm (LZMA), Prediction by Partial Matching (PPM), Burrows Wheeler Transform (BWT), Lempel Ziv Welch (LZW) coding, Deflate, LZ77 และ Deflate64 จากผลการวิจัย PPM, LZW และ Deflate64 ได้ผลการบีบอัดที่ดีกว่าวิธีอื่น ๆ

2.2.8 Uthayakumar และ Vengattaraman [11] ได้ทำวิจัยเพื่อเปรียบเทียบอัลกอริทึมการบีบอัดรูปภาพจากดาวเทียม เพื่อจัดการพื้นที่จัดเก็บและแบนด์วิดธ์การสื่อสาร แม้ว่าจะมีงานวิจัยหลายชิ้นเกี่ยวกับการบีบอัดภาพธรรมชาติ แต่มีเพียงไม่กี่งานที่เน้นภาพดาวเทียม ลักษณะของภาพถ่ายจากดาวเทียมก่อให้เกิดความท้าทายในการบีบอัดภาพดาวเทียม ในการวิจัยนี้ได้เลือกตัวอย่างรูปภาพจำนวน 2, 800 ภาพของเรือรบในภาพถ่ายดาวเทียม และได้เลือกใช้เทคนิคการบีบอัดภาพดังนี้

Lempel Ziv Markov (LZMA), Burrows Wheeler Transform (BWT), การเข้ารหัส Lempel Ziv Welch (LZW), Deflate และ LZ77 จากผลการทดลองปรากฏว่าเทคนิค LZMA สามารถทำการบีบอัดได้ดีกว่าวิธีอื่นด้วยอัตราส่วนการบีบอัดปัจจัยการบีบอัดและเวลาการบีบอัดที่ 0.5666

2.2.9 Preet และ Bagga [12] ได้ทำการทดสอบการประเมินผลการบีบอัดข้อมูลแบบไม่สูญเสีย โดยการใช้งานการบีบอัดข้อมูลผ่านระบบงานซึ่งมีการแลกเปลี่ยนข้อมูลกันผ่านโทรศัพท์มือถือ โดยใช้อัลกอริทึม lzop, gzip, lz4, bzip2, pzip2, pigz, xy จากผลการทดสอบประสิทธิภาพการบีบอัดอัลกอริทึม Lzop และ LZ4 มีอัตราความเร็วที่ดีทั้งการบีบอัดและการคลายข้อมูล

2.2.11 Xudong และ Yiran [4] ได้วิจัยเรื่องการบีบอัดข้อมูลแบบไม่สูญเสียข้อมูล โดยใช้การบีบอัดข้อมูลในรูปแบบของไฟล์ JSON สาเหตุมาจากการแลกเปลี่ยนข้อมูลเดิมในระบบอินเทอร์เน็ตในสรรพสิ่ง (Internet of Things) ซึ่งมีการแลกเปลี่ยนข้อมูลจากฝั่งไคลเอนต์และฝั่งเซิร์ฟเวอร์เดิมเป็นแบบ xml ซึ่งมีขนาดใหญ่ มาเป็นแบบ JSON พร้อมกับการบีบอัดข้อมูลแบบไม่สูญเสียข้อมูล จำนวน 4 อัลกอริทึมดังนี้ LZMA, GZIP, BZIP และ A-LZMA จากผลการวิจัย อัลกอริทึม BZIP มีประสิทธิภาพที่ดีกว่าวิธีอื่น ๆ

2.2.12 ชนาภา ศิลาวงษ์ และ ธนภัทร์ อนุศาสน์อมรกุล [1] ได้ทำการศึกษาเปรียบเทียบวิธีบีบอัดข้อมูลที่เหมาะสมสำหรับแต่ละประเภทข้อมูล ในงานวิจัยฉบับนี้ได้วิเคราะห์ประเภทข้อมูลหลายรูปแบบเพื่อหาอัลกอริทึมที่เหมาะสมสำหรับข้อมูลโดยรวม โดยวัดจากอัตราการบีบอัดข้อมูล เวลาที่ใช้ในการบีบอัดและคลายข้อมูล เวลารวมที่ใช้ในการรับ-ส่งข้อมูลผ่านเครือข่าย และอัตราการส่งข้อมูล ในงานวิจัยนี้ได้ศึกษาเปรียบเทียบอัลกอริทึมแบบไม่สูญเสีย 4 ประเภท ได้แก่ Huffman LZ77 LZW และ Deflate ผลลัพธ์แสดงให้เห็นว่า อัลกอริทึม Deflate มีประสิทธิภาพโดยรวมดีที่สุด

บทที่ 3

วิธีดำเนินการวิจัย

งานวิจัยนี้ดำเนินการวิจัยโดยมีวัตถุประสงค์ เพื่อเปรียบเทียบประสิทธิภาพอัลกอริทึมบีบอัดแบบไม่สูญเสียข้อมูลที่เหมาะสมกับหมวดไฟล์เอกสาร หมวดไฟล์รูปภาพ และหมวดไฟล์เพลงและวิดีโอ โดยวิธีการดำเนินงานวิจัยในครั้งนี้ ได้แบ่งขั้นตอนการทำงานออกเป็น 3 ขั้นตอน คือ 1) การเก็บรวบรวมไฟล์ข้อมูลและจัดการแยกไฟล์ตามหมวดหมู่ไฟล์ 2) การประมวลผลการบีบอัดข้อมูล/คลายข้อมูล การรับ-ส่งไฟล์ข้อมูล และการตรวจสอบความถูกต้องของไฟล์หลังการคลายข้อมูล 3) การประเมินประสิทธิภาพการทำงาน ซึ่งสามารถอธิบายรายละเอียดในแต่ละขั้นตอน ได้ดังนี้

เก็บรวบรวมไฟล์ข้อมูลและ
จัดการแยกไฟล์ตามหมวดหมู่ไฟล์

การบีบอัดข้อมูล/คลายข้อมูล/การรับ-ส่ง
ข้อมูล และการตรวจสอบความถูกต้อง
ของไฟล์หลังการคลายข้อมูล

ประเมินประสิทธิภาพ
การทำงาน

ภาพที่ 3.1 ขั้นตอนการดำเนินการวิจัย

3.1 การเก็บรวบรวมไฟล์ข้อมูลและจัดการแยกไฟล์ข้อมูลตามหมวดหมู่ขนาดไฟล์

ในการวิจัยครั้งนี้ ผู้วิจัยได้ทำการรวบรวมไฟล์ข้อมูลจากที่ต่าง ๆ จำนวน 150 ไฟล์ข้อมูล ซึ่งได้

คัดสรรไฟล์ตามหมวดหมู่ไฟล์ข้อมูล จำนวน 3 หมวดหมู่ คือ หมวดไฟล์เอกสาร หมวดไฟล์รูปภาพ และหมวดไฟล์เพลงและวิดีโอ ดังต่อไปนี้

3.1.1 หมวดไฟล์เอกสาร

ตารางที่ 3.1 ข้อมูลหมวดหมู่ไฟล์เอกสาร

ลำดับที่	ช่วงขนาดไฟล์ข้อมูล	จำนวนไฟล์ข้อมูล	ประเภทเอกสาร
1	ตั้งแต่ 1KB – 5KB	3	PDF, BAK XLS, XLSX, LOG, TXT, DOC, DOCX
2	ตั้งแต่ 5KB – 10KB	3	
3	ตั้งแต่ 10KB – 50KB	2	
4	ตั้งแต่ 50KB – 200KB	3	
5	ตั้งแต่ 200KB – 500KB	2	
6	ตั้งแต่ 500KB – 800KB	3	
7	ตั้งแต่ 800KB – 1000KB	3	
8	ตั้งแต่ 1MB – 5MB	10	
9	ตั้งแต่ 5MB – 10MB	10	
10	ตั้งแต่ 10MB – 50MB	10	
11	ตั้งแต่ 50MB – 60MB	3	
12	ตั้งแต่ 60MB – 70MB	2	
13	ตั้งแต่ 70MB – 80MB	2	
14	ตั้งแต่ 80MB – 100MB	2	
15	ตั้งแต่ 100MB ขึ้นไป	2	
รวม :		60	

พหุบัน ปณ กิจโต ชีเว

3.1.2 หมวดไฟล์รูปภาพ

ตารางที่ 3.2 ข้อมูลหมวดไฟล์รูปภาพ

ลำดับที่	ช่วงขนาดไฟล์ข้อมูล	จำนวนไฟล์ข้อมูล	ประเภทเอกสาร
1	ตั้งแต่ 1KB – 5KB	3	JPG, GIF, TIF, TIFF, PNG, BMP, PSD
2	ตั้งแต่ 5KB – 10KB	3	
3	ตั้งแต่ 10KB – 50KB	3	
4	ตั้งแต่ 50KB – 200KB	3	
5	ตั้งแต่ 200KB – 500KB	3	
6	ตั้งแต่ 500KB – 800KB	3	
7	ตั้งแต่ 800KB – 1000KB	5	
8	ตั้งแต่ 1MB – 5MB	6	
9	ตั้งแต่ 5MB – 10MB	6	
10	ตั้งแต่ 10MB – 50MB	3	
11	ตั้งแต่ 50MB – 60MB	2	
12	ตั้งแต่ 60MB – 70MB	2	
13	ตั้งแต่ 70MB – 80MB	2	
14	ตั้งแต่ 80MB – 100MB	2	
15	ตั้งแต่ 100MB ขึ้นไป	2	
รวม :		48	

พหุบัน ปณ กิโต ชีเว

3.1.3 หมวดไฟล์เพลงและวิดีโอ

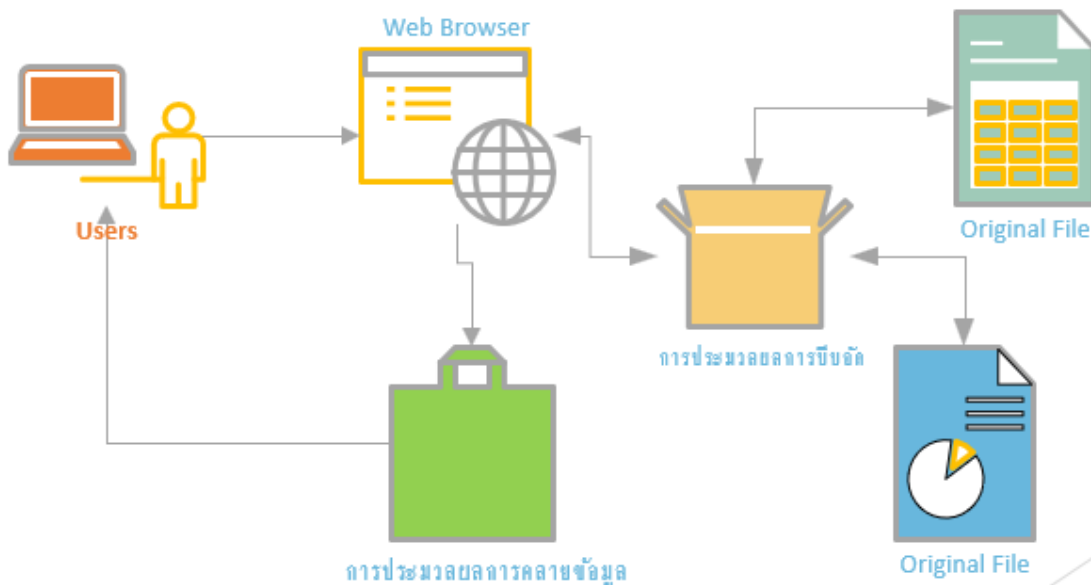
ตารางที่ 3.3 ข้อมูลหมวดไฟล์เพลงและวิดีโอ

ลำดับที่	ช่วงขนาดไฟล์ข้อมูล	จำนวนไฟล์ข้อมูล	ประเภทเอกสาร
1	ตั้งแต่ 1KB – 5KB	0	MP3, WAV, AVI, MP4, 3GP, WMA
2	ตั้งแต่ 5KB – 10KB	0	
3	ตั้งแต่ 10KB – 50KB	0	
4	ตั้งแต่ 50KB – 200KB	0	
5	ตั้งแต่ 200KB – 500KB	0	
6	ตั้งแต่ 500KB – 800KB	0	
7	ตั้งแต่ 800KB – 1000KB	0	
8	ตั้งแต่ 1MB – 5MB	5	
9	ตั้งแต่ 5MB – 10MB	5	
10	ตั้งแต่ 10MB – 50MB	6	
11	ตั้งแต่ 50MB – 60MB	6	
12	ตั้งแต่ 60MB – 70MB	5	
13	ตั้งแต่ 70MB – 80MB	5	
14	ตั้งแต่ 80MB – 100MB	5	
15	ตั้งแต่ 100MB ขึ้นไป	5	
รวม :		42	

พหุบัน ปณ กิจโต ชีเว

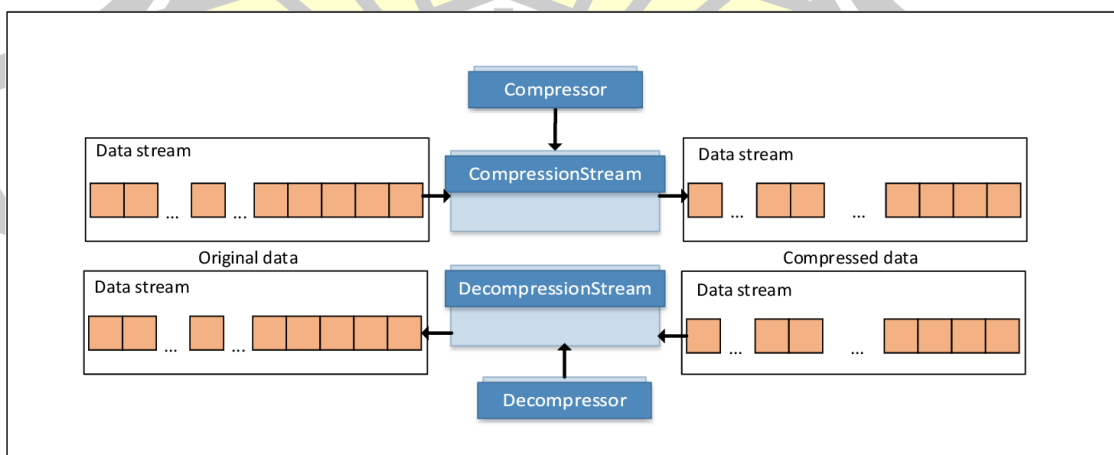
3.2 ประมวลผลการบีบอัด/คลายข้อมูล และการรับ-ส่งข้อมูล

สำหรับงานวิจัยนี้ จะออกแบบให้สามารถวัดประสิทธิภาพอัตราการบีบอัดข้อมูลรวมถึงอัตราความเร็วที่ใช้ในการบีบอัดและการคลายข้อมูลในแต่ละเทคนิค รวมถึงเวลารวมในการรับ-ส่งข้อมูล ซึ่งมีภาพรวมกระบวนการประมวลผล ในการทำงานดังภาพที่ 3.2



ภาพที่ 3.2 ภาพรวมการประมวลผลการบีบอัด/คลายข้อมูล

ในกระบวนการบีบอัด/คลายข้อมูล จะเป็นการนำเข้าไฟล์และทำการแปลงไฟล์ให้อยู่ในรูปแบบของ File Stream ซึ่งจะแบ่งข้อมูลออกเป็นบล็อกข้อมูลและจะทำการประมวลผลการบีบอัด/คลายข้อมูลตามแต่ละเทคนิค การแบ่งข้อมูลออกเป็นบล็อก ดังภาพที่ 3.3



ภาพที่ 3.3 การแบ่งข้อมูลออกเป็นบล็อก

ที่มา: <https://software.intel.com/sites/products/documentation/doclib/>

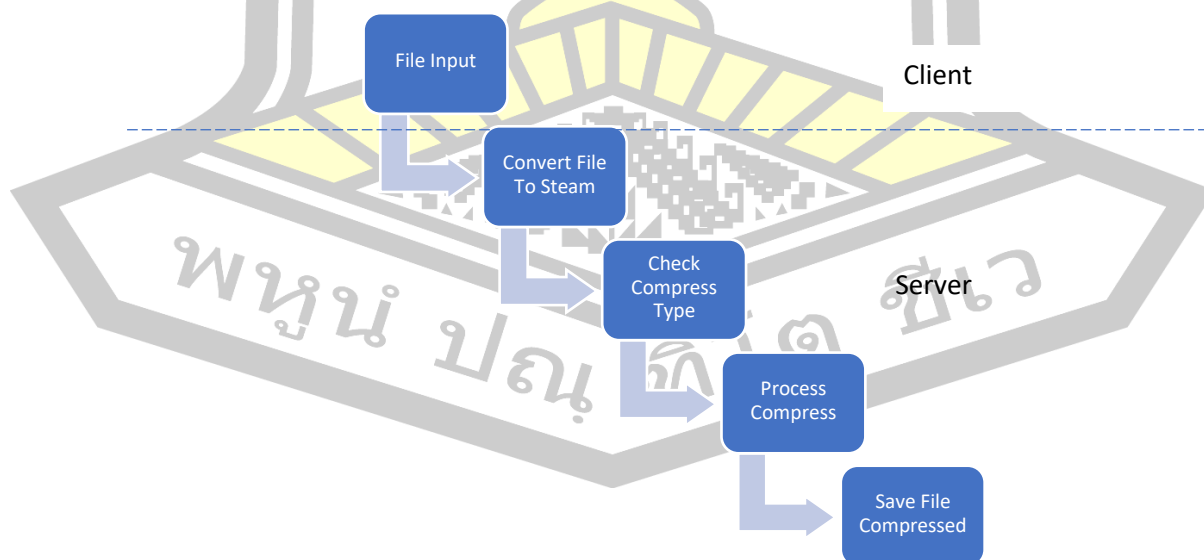
daal/daal-user-and-reference-guides/daal_prog_guide/GUID-5FC507AC-3AA9-424A-A520-1683637DD2ED.htm

3.2.1 กระบวนการบีบอัดข้อมูล

เป็นขั้นตอนเริ่มต้นจากผู้ใช้งานทำการเลือกไฟล์ที่ต้องการอัปโหลด เมื่อไฟล์ถูกอัปโหลดเข้ามาจัดเก็บที่เครื่องบริการเว็บเสร็จเรียบร้อยแล้ว จากนั้นจะทำการประมวลผลการบีบอัดข้อมูลตามอัลกอริทึมที่ผู้ใช้งานต้องการ เมื่อการประมวลผลการบีบอัดเสร็จสิ้น ไฟล์ข้อมูลจะถูกบันทึกจัดเก็บโดยที่นามสกุลไฟล์ข้อมูลจะต่อท้ายด้วยอัลกอริทึมที่ทำการบีบอัดเข้ามาด้วยทุกไฟล์ข้อมูล ในขั้นตอนการบีบอัดข้อมูล (Compression) จะประมวลผลการทำงานที่เครื่องบริการเว็บ ในกระบวนการบีบอัดข้อมูลจะทำการแยกนามสกุลไฟล์เอกสารตามอัลกอริทึม ดังนี้

- HuffmanCodes จะใช้นามสกุลเป็น .hmc
- BZip2 จะใช้นามสกุลเป็น .bz2
- DEFLATE จะใช้นามสกุลเป็น .df
- LZMA จะใช้นามสกุลเป็น .lzma
- LZ4 จะใช้นามสกุลเป็น .lz4

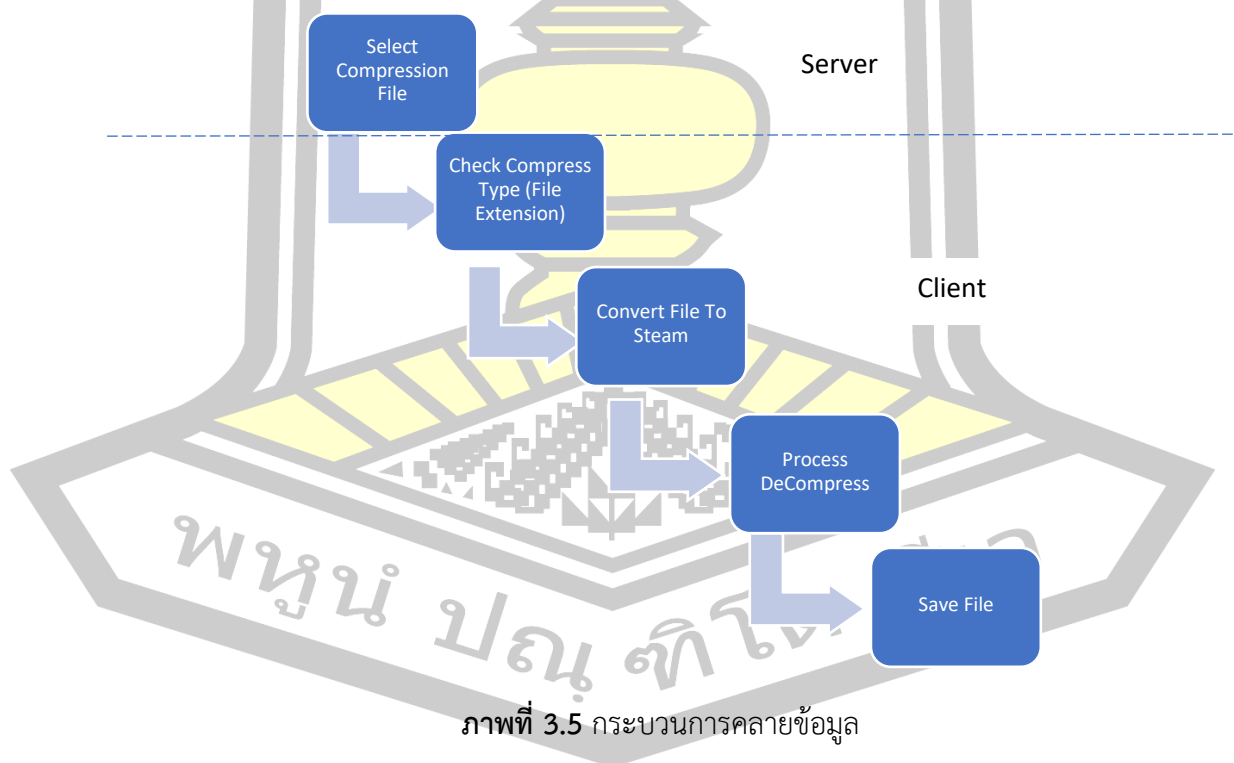
ซึ่งขั้นตอนการบีบอัดข้อมูลจะมีกระบวนการทำงานดังต่อไปนี้



ภาพที่ 3.4 กระบวนการบีบอัดข้อมูล

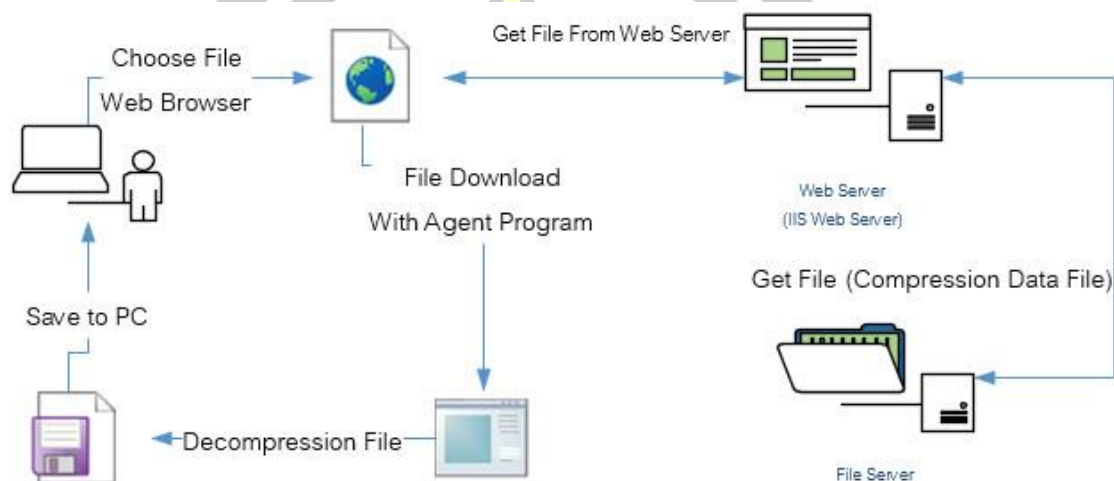
3.2.2 กระบวนการคลายข้อมูล

เป็นขั้นตอนในการคลายข้อมูลที่มีการบีบอัดมาแล้ว โดยเริ่มต้นการทำงานหลังจากโปรแกรมดาวน์โหลดไฟล์ จากเครื่องบริการเว็บลงมาจัดเก็บที่เครื่องผู้ใช้งานก่อน จากนั้นโปรแกรมทำการคลายการบีบอัดข้อมูล โดยการตรวจสอบนามสกุลไฟล์ก่อน เนื่องจากนามสกุลไฟล์เอกสารที่ทำการบีบอัดข้อมูลมาแล้วจะบ่งบอกชนิดของอัลกอริทึมที่ใช้ในการบีบอัด เมื่อบีบอัดด้วยอัลกอริทึมใด ๆ ต้องทำการคลายการบีบอัดด้วยอัลกอริทึมนั้น ๆ เสมอ ในขั้นตอนการคลายข้อมูลจะประมวลผลการทำงานที่เครื่องผู้ใช้งาน (Client) การประมวลผลการทำงานที่เครื่องผู้ใช้งาน จะทำให้มีประสิทธิภาพมากกว่าการประมวลผลการทำงานที่เครื่องบริการเว็บ เนื่องจากในกรณีที่ผู้ใช้จำนวนมากต้องการดาวน์โหลดไฟล์เอกสารไปใช้งานพร้อม ๆ กัน ส่งผลให้เครื่องบริการเว็บต้องประมวลผลข้อมูลในหลาย ๆ ขั้นตอน ซึ่งอาจจะส่งผลให้ประสิทธิภาพการทำงานลดลง ดังนั้นในการวิจัยครั้งนี้จึงได้เลือกใช้เทคนิคในการคลายข้อมูลที่เครื่องผู้ใช้งาน เพราะจะทำให้การทำงานของเครื่องบริการเว็บ มีประสิทธิภาพมากยิ่งขึ้น ซึ่งขั้นตอนการคลายข้อมูลจะมีกระบวนการทำงานดังภาพที่ 3.4



3.2.3 กระบวนการรับ-ส่งข้อมูลผ่านเว็บแอปพลิเคชัน

ในการวิจัยครั้งนี้ ได้ทำการเลือกโอนถ่ายข้อมูลแบบ Web Client ซึ่งรูปแบบของ Web Client จะเป็นการสร้างโปรแกรม (Agent Program) สำหรับการจัดการการดาวน์โหลดไฟล์จากเว็บไซต์ ดังภาพที่ 3.6



ภาพที่ 3.6 กระบวนการรับ-ส่งไฟล์ และการคลายข้อมูล

จากภาพที่ 3.6 แสดงขั้นตอนในการถ่ายโอนข้อมูลจากเว็บแอปพลิเคชันไปที่เครื่องผู้ใช้งาน โดยเริ่มต้นการทำงานจากผู้ใช้งานทำการคลิกเลือกไฟล์ข้อมูลที่ต้องการดาวน์โหลด จากรายการไฟล์ที่แสดงบนเว็บแอปพลิเคชัน จากนั้นโปรแกรมจะทำการส่งค่าพารามิเตอร์ (URL) ให้กับโปรแกรมดาวน์โหลดไฟล์เอกสาร (API) เมื่อโปรแกรมทำการดาวน์โหลดไฟล์จากเครื่องบริการเว็บลงมาถึงเก็บที่เครื่องผู้ใช้งานเสร็จเรียบร้อย โปรแกรมจะทำการคลายการบีบอัดข้อมูล และบันทึกจัดเก็บไฟล์ข้อมูลที่เครื่องผู้ใช้งาน

3.2.4 การตรวจสอบความถูกต้องของไฟล์หลังการคลายข้อมูล

เป็นขั้นตอนในการตรวจสอบไฟล์ต้นฉบับก่อนการบีบอัดไฟล์กับไฟล์หลังจากการคลายข้อมูล เพื่อเปรียบเทียบไฟล์ทั้ง 2 ไฟล์ไม่ให้เกิดความสูญเสียข้อมูลเกิดขึ้น ในงานวิจัยครั้งนี้ ผู้วิจัยได้ทำการเลือกใช้อัลกอริทึมการบีบอัดแบบไม่สูญเสียข้อมูล ดังนั้นการตรวจสอบไฟล์จึงมีความสำคัญว่าทั้ง 2 ไฟล์เมื่อทำการบีบอัดและคลายข้อมูลออกมาแล้ว ข้อมูลจะต้องไม่เกิดความสูญเสียเกิดขึ้น ซึ่งขั้นตอนนี้จะใช้หลักการตรวจสอบไฟล์จำนวน 2 ขั้นตอนดังนี้ 1) ทำการเปรียบเทียบขนาดไฟล์ข้อมูล 2 ไฟล์ คือไฟล์ต้นฉบับ กับไฟล์หลังการบีบอัด ถ้าขนาดไฟล์แตกต่างกัน โปรแกรมดาวน์โหลดไฟล์จะทำการลบไฟล์ที่ดาวน์โหลดมาก่อนหน้าทิ้ง แล้วทำขั้นตอนในการ รับ-ส่งไฟล์ใหม่อีกครั้ง และทำการ

ตรวจสอบจนกว่าขนาดไฟล์ทั้ง 2 ไฟล์จะมีเนื้อหาที่เท่ากัน เมื่อขนาดไฟล์มีค่าเท่ากัน จะทำขั้นตอนที่ 2 ต่อไป 2) ขั้นตอนนี้จะขั้นตอนในการอ่านค่าไฟล์ทั้ง 2 ไฟล์ออกมาอยู่ในรูปแบบของ Byte ก่อน จากนั้นจะทำการเปรียบเทียบข้อมูลทั้ง 2 ไฟล์เป็นรายบรรทัด โดยเริ่มต้นที่บรรทัดที่ 1 จนถึงบรรทัดสุดท้ายของไฟล์ข้อมูลทั้ง 2 โดยที่แต่ละรอบในการตรวจสอบไฟล์ ถ้าบรรทัดไหนทำการตรวจสอบแล้ว ข้อมูลไม่ตรงกัน จะหยุดการทำงาน และแจ้งผลเป็น “Files are not equal.” แต่ถ้าทำการตรวจสอบแล้วไฟล์เหมือนกันทุกบรรทัด จะแจ้งผลเป็น “Files are equal.” จากขั้นตอนการทำงานทั้ง 2 ขั้นตอนนี้สามารถเขียนเป็นโปรแกรมเพื่อใช้ในการตรวจสอบไฟล์ข้อมูลได้ ดังนี้

ตัวอย่างโค้ดเปรียบเทียบไฟล์

```
// This method accepts two strings the represent two files to
// compare. A return value of 0 indicates that the contents
// of the files
// are the same. A return value of any other value indicates
// that the
// files are not the same.
private bool FileCompare(string file1, string file2)
{
    int file1byte;
    int file2byte;
    FileStream fs1;
    FileStream fs2;

    // Determine if the same file was referenced two times.
    if (file1 == file2)
    {
        // Return true to indicate that the files are the
same.
        return true;
    }

    // Open the two files.
    fs1 = new FileStream(file1, FileMode.Open);
    fs2 = new FileStream(file2, FileMode.Open);

    // Check the file sizes. If they are not the same, the
files
    // are not the same.
    if (fs1.Length != fs2.Length)
    {
        // Close the file
        fs1.Close();
        fs2.Close();
    }
}
```

```

        // Return false to indicate files are different
        return false;
    }

    // Read and compare a byte from each file until either a
    // non-matching set of bytes is found or until the end
of
    // file1 is reached.
    do
    {
        // Read one byte from each file.
        file1byte = fs1.ReadByte();
        file2byte = fs2.ReadByte();
    }
    while ((file1byte == file2byte) && (file1byte != -1));

    // Close the files.
    fs1.Close();
    fs2.Close();

    // Return the success of the comparison. "file1byte" is
    // equal to "file2byte" at this point only if the files
are
    // the same.
    return ((file1byte - file2byte) == 0);
}
private void button1_Click(object sender, System.EventArgs e)
{
    // Compare the two files that referenced in the textbox
controls.
    if (FileCompare(this.txtFile1.Text, this.txtFile2.Text))
    {
        MessageBox.Show("Files are equal.");
    }
    else
    {
        MessageBox.Show("Files are not equal.");
    }
}
}

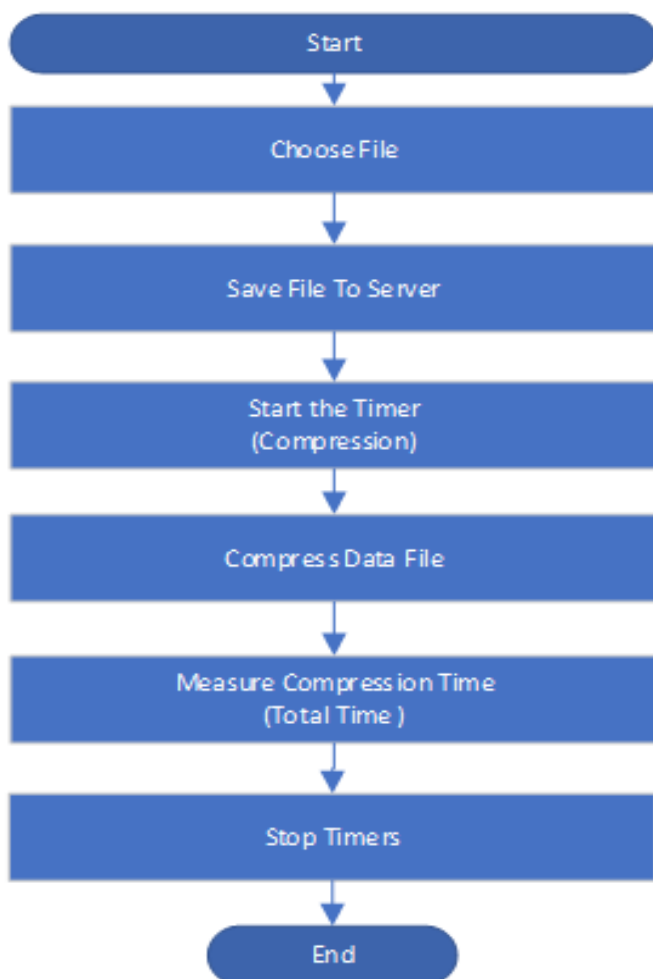
```

3.3 การประเมินประสิทธิภาพการทำงาน

ในการวิจัยครั้งนี้ ผู้วิจัยจะประเมินประสิทธิภาพการทำงานของการบีบอัดไฟล์ ซึ่งจะแบ่งการทำงานออกเป็น 2 กระบวนการ คือ 1) กระบวนการอัปโหลดไฟล์เข้าระบบและทำการบีบอัดข้อมูล เมื่อไฟล์ถูกอัปโหลดเข้าระบบเสร็จเรียบร้อยแล้ว จะทำการเริ่มการบีบอัดไฟล์และวัดประสิทธิภาพการบีบอัด ดังภาพที่ 3.6 2) กระบวนการดาวน์โหลดไฟล์และทำการคลายข้อมูล เมื่อมีการคลิกดาวน์โหลด

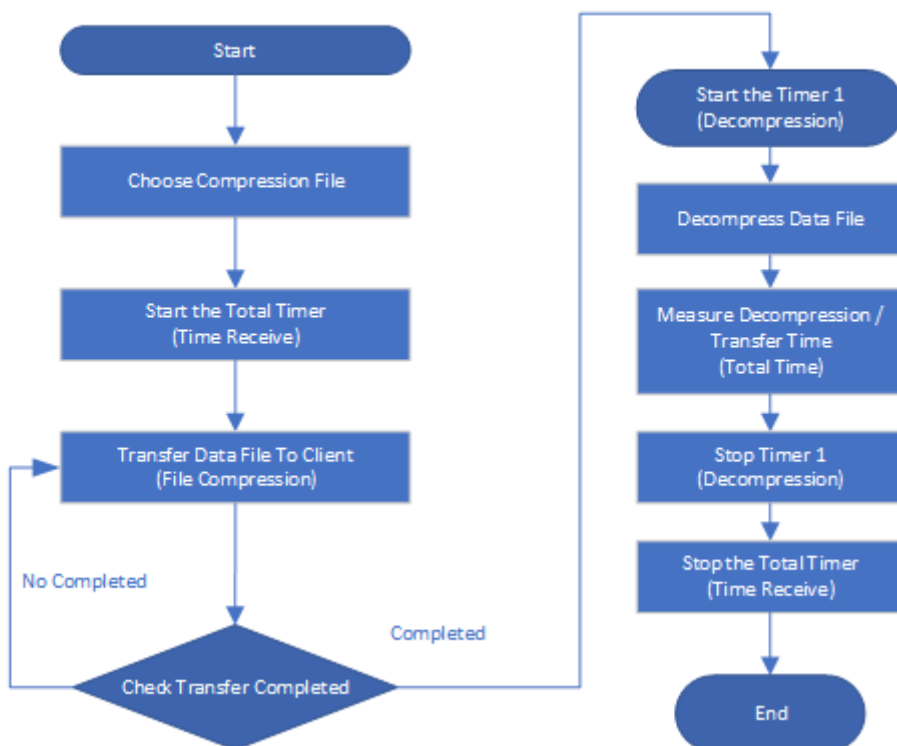
ไฟล์จากเว็บแอปพลิเคชัน ระบบส่งลิงค์ดาวน์โหลดให้กับ API เพื่อทำการดาวน์โหลดไฟล์จาก Web Server ลงมาเก็บที่เครื่อง Client หลักจากไฟล์ดาวน์โหลดเสร็จสิ้น ตัว API จะทำการคลายข้อมูลให้อัตโนมัติ ซึ่งขั้นตอนนี้จะวัดประสิทธิภาพการคลายข้อมูลและเวลารวมในการรับส่งไฟล์ตั้งแต่เริ่มดาวน์โหลดไฟล์จนถึงสิ้นสุดกระบวนการคลายข้อมูล ดังภาพที่ 3.7

กระบวนการอัปโหลดไฟล์เข้าระบบ (Compression File)



ภาพที่ 3.7 ขั้นตอนการอัปโหลดและบีบอัดไฟล์

กระบวนการดาวน์โหลดไฟล์และคลายข้อมูล (Decompression File)



ภาพที่ 3.8 ขั้นตอนการดาวน์โหลดและคลายข้อมูล

ในการวิจัยนี้ ผู้วิจัยจะทำการวัดประสิทธิภาพการทำงานจำนวน 4 ตัวชี้วัด ดังนี้

1. การวัดอัตราความเร็วการบีบอัดข้อมูล (Compression Speed) [11-13] คืออัตราเร็วในการบีบอัดข้อมูลซึ่งเป็นอัตราส่วนระหว่างปริมาณข้อมูลและ เวลาที่ใช้ในการบีบอัดข้อมูล มีหน่วยเป็นไบต์ต่อวินาที ดังสมการ (3-1)

$$\text{Compression speed} = \frac{\text{file size}}{\text{compression time}} \quad (3-1)$$

- File size คือ ขนาดข้อมูลก่อนการบีบอัดมีหน่วยเป็น ไบต์

- Compression Time คือ เวลาที่ใช้ในการบีบอัด ข้อมูล มีหน่วยเป็นวินาที

2. การวัดอัตราความเร็วการคลายข้อมูล (Decompression Speed) [11-13] คืออัตราเร็วในการ คลายข้อมูล ซึ่งเป็นอัตราส่วนระหว่างปริมาณข้อมูล และเวลาที่ใช้ในการคลายข้อมูล มีหน่วยเป็นไบต์ต่อ วินาที ดังสมการที่ (3-2)

$$\text{Decompression speed} = \frac{\text{file size1}}{\text{decompression time}} \quad (3-2)$$

- File size1 คือขนาดข้อมูลหลังการบีบอัด มีหน่วยเป็นไบต์
- Decompression Time คือ เวลาที่ใช้ในการคลายข้อมูลมีหน่วยเป็นวินาที

3. เวลารวมทั้งหมดทุกกระบวนการ (Overall Time) [14, 15] ตั้งแต่เริ่มบีบอัดข้อมูล จนถึงกระบวนการคลายข้อมูลจากผู้ร้องขอไฟล์ข้อมูลผ่านเว็บแอปพลิเคชัน ดังสมการที่ 3-3

$$\begin{aligned} \text{Overall Times} &= ((\text{Compression Times} \\ &+ \text{Processing Times}) \\ &+ (\text{Decompression Times} \\ &+ \text{Transmission Times})) \end{aligned} \quad (3-3)$$

- Transmission Times คือเวลาที่ใช้ในการส่งข้อมูลจากต้นทางไปถึงปลายทางที่ร้องขอ
- Processing Times คือเวลาที่ใช้ในการประมวลผลการทำงาน
- Compression Times คือเวลาที่ใช้ในการบีบอัดข้อมูล
- Decompression Times คือเวลาที่ใช้ในการคลายข้อมูล

4. ประสิทธิภาพการบีบอัดข้อมูลจะวัดจาก อัตราส่วนการบีบอัดข้อมูล [14, 15] ซึ่งแสดงถึงจำนวนข้อมูลที่ เหลือหลังจากการบีบอัด ความสัมพันธ์ดังกล่าวจะอยู่ในรูปของร้อยละขนาดของข้อมูลดั้งเดิม (File size2) และขนาดของข้อมูลที่ถูกบีบอัดแล้ว (Compression size) ดังสมการที่ (3-4)

$$\text{Compression ratio} = \frac{\text{Compression size}}{\text{file size2}} * 100 \quad (3-4)$$

พหุ ประสิทธิภาพ

บทที่ 4

ผลการดำเนินงานวิจัย

ผลการดำเนินการวิจัยในการเปรียบเทียบประสิทธิภาพการบีบอัดแบบไม่สูญเสียข้อมูลบนเว็บแอปพลิเคชันและนำผลการเปรียบเทียบที่มีประสิทธิภาพด้านความเร็วมากที่สุดมาสร้างต้นแบบการบีบอัดข้อมูลบนเว็บแอปพลิเคชัน ในการทดสอบวิจัยครั้งนี้ ผู้วิจัยได้ใช้ไฟล์ข้อมูลในการทดสอบจำนวนทั้งหมด 150 ไฟล์ข้อมูล โดยการจัดแยกไฟล์ตามหมวดหมู่ไฟล์ ดังนี้ 1) หมวดไฟล์เอกสาร จำนวน 60 ไฟล์ 2) หมวดไฟล์รูปภาพ จำนวน 48 ไฟล์ 3) หมวดไฟล์เพลงและวิดีโอ จำนวน 42 ไฟล์ และแบ่งช่วงกลุ่มขนาดไฟล์ออกเป็น 3 ช่วงกลุ่ม ประกอบไปด้วย กลุ่มไฟล์ขนาดเล็ก กลุ่มไฟล์ขนาดกลาง และกลุ่มไฟล์ขนาดใหญ่ จากนั้นทำการทดสอบการบีบอัดแบบไม่สูญเสียข้อมูลจำนวน 5 อัลกอริทึม คือ Huffman Coding, LZMA, LZ4, DEFLATE และ BZip2 และทำการวัดประสิทธิภาพการทำงานด้วยการวัดอัตราความเร็วในการบีบอัด อัตราความเร็วในการคลายข้อมูล อัตราส่วนการบีบอัด เวลาการทำงานทุกกระบวนการ และนำผลการทดสอบมาสร้างต้นแบบการบีบอัดข้อมูลผ่านเว็บแอปพลิเคชัน จากการทดสอบสามารถแสดงผลการเปรียบเทียบและผลการสร้างต้นแบบการบีบอัด ได้ดังนี้

4.1 ผลการเปรียบเทียบอัตราความเร็วการบีบอัด (Compression speed)

เป็นการวัดหาอัตราความเร็วที่ใช้ในการบีบอัดข้อมูล จากเริ่มต้นจนถึงสิ้นสุดการบีบอัดข้อมูล ซึ่งผลลัพธ์ที่ได้จะมีหน่วยเป็นไบต์ต่อวินาที โดยสามารถแสดงผลการทดสอบได้ดังตารางที่ 4.1

ตารางที่ 4.1 อัตราความเร็วในการบีบอัดข้อมูล

อัลกอริทึม	File Group (byte / s)			
	Music & Videos	Images	Data	ค่าเฉลี่ยรวม
LZ4	133, 034, 743.74	65, 835, 663.25	42, 744, 585.52	77, 262, 260.91
DEFLATE	23, 073, 507.59	32, 384, 160.89	20, 504, 606.93	25, 975, 720.70
Huffman Coding	2, 371, 039.79	2, 303, 321.63	2, 120, 797.02	2, 263, 874.84
BZip2	1, 023, 002.04	1, 608, 261.90	1, 408, 694.51	1, 380, 527.57
LZMA	830, 718.02	840, 928.95	783, 964.60	820, 082.08

จากตารางที่ 4.1 แสดงผลลัพธ์การวัดอัตราความเร็วในการบีบอัดไฟล์ข้อมูล จากผลการทดลองพบว่าอัลกอริทึม LZ4 สามารถทำความเร็วในการบีบอัดข้อมูลดีที่สุด ที่ความเร็วเฉลี่ย 77, 262, 260.91 ไบต์ต่อวินาที และยังพบว่าอัลกอริทึม LZ4 สามารถทำความเร็วได้ดีทั้ง 3 หมวดหมู่

ไฟล์ โดยที่หมวดไฟล์เพลงและวิดีโอ มีความเร็วเฉลี่ย 133, 034, 743.74 ไบต์ต่อวินาที หมวดไฟล์เอกสาร ความเร็วเฉลี่ย 65, 835, 663.25 ไบต์ต่อวินาที หมวดไฟล์รูปภาพ ความเร็วเฉลี่ย 42, 744, 585.52 ไบต์ต่อวินาที จะเห็นได้ว่า อัลกอริทึม LZ4 จะมีอัตราความเร็วในการบีบอัดที่สูงกว่า อัลกอริทึมอื่น ๆ เป็นอย่างมาก โดยมีค่าเฉลี่ยมากกว่า อัลกอริทึม DEFLATE ที่มีลำดับถัดมา ที่มีความเร็วเฉลี่ย 51, 286, 540.21 ไบต์ต่อวินาที

ตารางที่ 4.2 อัตราความเร็วในการบีบอัดข้อมูล แยกตามช่วงขนาดไฟล์ข้อมูล

ช่วงขนาดไฟล์	File Group (byte / s)				
	BZip2	DEFLATE	Huffman Coding	LZ4	LZMA
หมวดหมู่ไฟล์: Group_Data					
1KB – 5KB	241, 634.34	949, 786.32	197, 665.72	814, 585.96	95, 974.89
5KB – 10KB	650, 647.02	2, 417, 065.52	663, 157.56	2, 318, 376.67	277, 085.66
10KB – 50KB	1, 081, 954.01	4, 063, 316.71	1, 358, 452.11	4, 081, 285.15	579, 264.81
50KB – 200KB	1, 048, 096.68	10, 484, 120.69	2, 132, 819.72	12, 398, 439.55	768, 609.04
200KB – 500KB	1, 586, 217.85	19, 981, 230.40	2, 537, 779.47	22, 321, 464.05	915, 254.34
500KB – 800KB	996, 166.55	19, 884, 225.81	2, 388, 155.96	43, 041, 368.83	975, 060.51
800KB – 1000KB	1, 601, 320.74	38, 436, 064.25	2, 683, 316.83	52, 201, 731.11	1, 121, 347.74
1MB – 5MB	1, 488, 553.42	31, 906, 316.66	2, 738, 196.31	51, 137, 569.70	1, 009, 126.88
5MB – 10MB	4, 005, 333.52	84, 811, 795.79	2, 721, 989.00	87, 833, 147.68	1, 167, 031.59
10MB – 50MB	1, 215, 316.20	25, 837, 849.66	2, 472, 749.03	74, 883, 913.20	783, 331.68
50MB – 60MB	866, 529.85	22, 447, 625.55	2, 419, 607.26	145, 806, 714.11	742, 888.27
60MB – 70MB	1, 002, 140.79	23, 411, 172.55	2, 452, 207.22	103, 443, 732.84	689, 971.32
70MB – 80MB	862, 338.14	22, 875, 090.65	2, 491, 739.23	181, 284, 602.62	806, 441.02
80MB – 100MB	1, 025, 879.69	23, 295, 599.60	2, 440, 111.95	101, 409, 763.69	734, 707.70
100MB ขึ้นไป	1, 036, 717.98	23, 190, 274.21	2, 427, 602.69	108, 384, 071.87	733, 329.30
หมวดหมู่ไฟล์: Group_Images					
1KB – 5KB	190, 602.65	475, 377.77	109, 285.26	543, 318.49	72, 956.99
5KB – 10KB	439, 773.59	1, 139, 095.61	529, 929.54	2, 134, 857.28	281, 202.08
10KB – 50KB	1, 418, 623.37	5, 760, 704.03	1, 452, 807.45	4, 041, 681.57	706, 493.00
50KB – 200KB	1, 179, 303.34	11, 433, 760.57	1, 941, 759.82	12, 884, 321.06	1,020, 431.29
200KB – 500KB	1, 018, 055.50	15, 305, 872.19	2, 233, 781.39	29, 254, 656.96	965, 690.30
500KB – 800KB	1, 060, 311.34	20, 562, 020.92	2, 391, 186.73	67, 363, 914.61	1,024, 398.75
800KB – 1000KB	1, 228, 490.16	22, 194, 041.38	2, 308, 553.66	63, 798, 166.97	1,012, 760.13
1MB – 5MB	2, 267, 006.60	37, 094, 283.12	2, 774, 691.67	61, 619, 465.17	1,243, 953.19
5MB – 10MB	1, 319, 706.70	23, 815, 810.38	2, 397, 411.77	101, 770, 107.05	783, 410.56
10MB – 50MB	1, 491, 037.76	24, 241, 088.76	2, 467, 033.83	56, 346, 812.04	731, 304.30

ตารางที่ 4.2 อัตราความเร็วในการบีบอัดข้อมูล แยกตามช่วงขนาดไฟล์ข้อมูล

ช่วงขนาดไฟล์	File Group (byte / s)				
	BZip2	DEFLATE	Huffman Coding	LZ4	LZMA
50MB – 60MB	1, 784, 407.73	24, 541, 784.95	2, 483, 350.81	19, 783, 807.86	518, 370.36
60MB – 70MB	1, 793, 274.33	25, 065, 378.80	2, 422, 407.65	20, 929, 564.67	533, 996.63
70MB – 80MB	1, 747, 368.79	22, 969, 264.24	2, 529, 916.53	20, 846, 899.76	568, 799.54
80MB – 100MB	2, 660, 530.21	38, 856, 571.24	2, 968, 580.70	39, 393, 035.32	809, 417.47
100MB ขึ้นไป	1, 795, 160.48	24, 085, 303.36	2, 518, 502.33	16, 398, 267.83	459, 277.17
หมวดหมู่ไฟล์: Group_Music					
1MB – 5MB	1, 021, 109.47	23, 218, 798.13	2, 326, 530.98	106, 047, 084.81	934, 384.36
5MB – 10MB	1, 039, 835.21	23, 320, 249.27	2, 438, 624.05	95, 977, 529.92	910, 866.09
10MB – 50MB	975, 892.39	23, 084, 339.25	2, 441, 016.96	91, 560, 999.93	819, 344.64
50MB – 60MB	1, 011, 172.29	22, 411, 796.85	2, 364, 665.49	168, 780, 558.68	821, 028.48
60MB – 70MB	1, 078, 754.45	23, 657, 679.89	2, 266, 473.09	126, 614, 234.05	801, 808.27
70MB – 80MB	1, 059, 214.12	24, 013, 583.75	2, 324, 530.84	164, 131, 609.76	806, 001.12
80MB – 100MB	1, 039, 963.37	23, 143, 799.24	2, 375, 987.96	176, 784, 122.11	799, 201.15
100MB ขึ้นไป	969, 862.93	21, 867, 990.14	2, 417, 768.34	135, 527, 396.38	757, 322.62
ค่าเฉลี่ยรวม	1, 380, 527.57	25, 975, 720.70	2, 263, 874.84	77, 262, 260.91	820, 082.08

จากผลการทดสอบในตารางที่ 4.2 แสดงอัตราความเร็วในการบีบอัดข้อมูล โดยทำการแบ่งแยกตามช่วงขนาดไฟล์ 15 ช่วงไฟล์ พบว่าไฟล์ขนาด 1 กิโลไบต์ – 50 กิโลไบต์ อัลกอริทึม DEFLATE สามารถทำความเร็วในการบีบอัดได้ดีในหมวดไฟล์รูปภาพและหมวดไฟล์เอกสาร และอัลกอริทึม DEFLATE ยังสามารถทำความเร็วได้ดีในช่วงขนาดไฟล์ 50 เมกกะไบต์ – 50 เมกกะไบต์ ในหมวดไฟล์รูปภาพ แต่ค่าเฉลี่ยรวมกลับพบว่าอัลกอริทึม LZ4 มีอัตราความเร็วในการบีบอัดที่สูงกว่าอัลกอริทึมอื่น ๆ

4.2 ผลการเปรียบเทียบอัตราความเร็วการคลายข้อมูล (Decompression speed)

เป็นการวัดหาอัตราความเร็วที่ใช้ในการคลายข้อมูล จากเริ่มต้นจนถึงสิ้นสุดการคลายข้อมูล ซึ่งผลลัพธ์ที่ได้จะมีหน่วยเป็นไบต์ต่อวินาที โดยสามารถแสดงผลการทดสอบได้ดังตารางที่ 4.3

ตารางที่ 4.3 อัตราความเร็วในการคลายข้อมูล

อัลกอริทึม	File Group (byte / s)			
	Music & Videos	Images	Data	ค่าเฉลี่ยรวม
LZ4	763, 951, 843.31	104, 543, 522.90	297, 746, 985.22	366, 459, 237.54
DEFLATE	162, 223, 009.27	51, 296, 907.76	87, 952, 600.45	97, 018, 493.26
BZip2	3, 541, 681.15	1, 291, 306.45	2, 569, 048.82	2, 432, 508.32
LZMA	3, 022, 228.02	1, 607, 089.16	2, 382, 631.09	2, 318, 286.13
Huffman Coding	161, 432.40	83, 929.44	90, 452.61	108, 239.54

จากตารางที่ 4.3 แสดงผลลัพธ์การวัดอัตราความเร็วในการคลายข้อมูล จากผลการทดลองพบว่า อัลกอริทึม LZ4 สามารถทำความเร็วในการคลายข้อมูลดีที่สุด ที่ความเร็วเฉลี่ย 366,459,237.54 ไบต์ต่อวินาที และยังพบว่าอัลกอริทึม LZ4 สามารถทำความเร็วได้ดีทั้ง 3 หมวดหมู่ไฟล์ โดยที่หมวดหมู่ไฟล์เพลงและวิดีโอ มีความเร็วเฉลี่ย 763,951,843.31 ไบต์ต่อวินาที หมวดไฟล์เอกสาร ความเร็วเฉลี่ย 297,746,985.22 ไบต์ต่อวินาที หมวดไฟล์รูปภาพความเร็วเฉลี่ย 104,543,522.90 ไบต์ต่อวินาที และจากผลการทดสอบในตารางที่ 4.1 และตารางที่ 4.3 จะเห็นได้ว่า อัลกอริทึม LZ4 จะมีอัตราความเร็วดีที่สุด และมีอัตราความเร็วในการคลายข้อมูลที่สูงกว่าอัลกอริทึมอื่น ๆ เป็นอย่างมาก โดยมีค่าเฉลี่ยมากกว่า อัลกอริทึม DEFLATE ที่มีลำดับถัดมา ที่ความเร็วเฉลี่ย 269,440,744.28 ไบต์ต่อวินาที เนื่องจากอัลกอริทึม LZ4 ถูกพัฒนาและออกแบบให้มีความสามารถในการประมวลผลการบีบอัดและคลายข้อมูลที่เน้นอัตราความเร็วในการทำงานเป็นหลัก จึงส่งผลทำให้ผลการทดลองพบว่า อัลกอริทึม LZ4 มีอัตราความเร็วเฉลี่ยที่ดีที่สุด ทั้งในด้านการบีบอัดและด้านการคลายข้อมูล

ตารางที่ 4.4 อัตราความเร็วในการคลายข้อมูล แยกตามช่วงขนาดไฟล์ข้อมูล

ช่วงขนาดไฟล์	File Group (byte / s)				
	BZip2	DEFLATE	Huffman Coding	LZ4	LZMA
หมวดหมู่ไฟล์: Group_Data					
1KB – 5KB	228, 106.71	279, 437.62	205, 708.60	398, 710.87	219, 419.52
5KB – 10KB	606, 887.38	1, 541, 478.64	121, 220.99	2, 752, 933.38	619, 800.53
10KB – 50KB	990, 698.27	3, 408, 721.95	99, 176.61	4, 149, 379.73	957, 741.45
50KB – 200KB	2, 667, 768.45	24, 169, 632.33	91, 983.56	30, 317, 598.05	2, 240, 783.85
200KB – 500KB	2, 604, 989.28	31, 066, 124.37	90, 212.18	47, 626, 912.47	2, 394, 153.92
500KB – 800KB	3, 406, 397.57	53, 717, 477.34	87, 499.80	117, 507, 137.37	2, 857, 469.26
800KB – 1000KB	2, 222, 341.18	40, 926, 350.81	83, 712.91	98, 058, 249.05	2, 285, 814.61

ตารางที่ 4.4 อัตราความเร็วในการคลายข้อมูล แยกตามช่วงขนาดไฟล์ข้อมูล

ช่วงขนาดไฟล์	File Group (byte / s)				
	BZip2	DEFLATE	Huffman Coding	LZ4	LZMA
1MB – 5MB	2, 769, 142.26	51, 592, 254.06	79, 470.95	167, 447, 243.72	2, 588, 058.62
5MB – 10MB	2, 616, 711.60	62, 321, 712.64	76, 604.26	220, 166, 266.00	2, 452, 297.83
10MB – 50MB	3, 012, 883.50	115, 557, 501.19	84, 647.34	461, 737, 690.80	2, 843, 796.60
50MB – 60MB	3, 424, 251.42	239, 847, 771.51	78, 886.36	751, 234, 942.73	2, 851, 094.71
60MB – 70MB	2, 879, 509.14	196, 291, 042.21	71, 448.48	634, 376, 195.82	2, 745, 207.89
70MB – 80MB	3, 388, 790.65	380, 147, 346.59	76, 969.01	908, 853, 599.01	3, 095, 937.49
80MB – 100MB	3, 196, 516.97	230, 051, 673.22	81, 692.14	767, 935, 129.10	3, 157, 292.58
100MB ขึ้นไป	3, 183, 644.50	109, 532, 543.41	86, 948.71	822, 307, 980.63	3, 096, 260.42
หมวดหมู่ไฟล์: Group_Images					
1KB – 5KB	570, 154.19	444, 587.90	192, 783.78	400, 774.44	310, 734.58
5KB – 10KB	1, 069, 864.31	2, 675, 140.85	85, 443.67	2, 619, 972.33	1, 043, 304.65
10KB – 50KB	1, 063, 406.67	5, 648, 814.96	81, 647.55	6, 727, 332.07	1, 048, 672.10
50KB – 200KB	1, 604, 297.42	21, 750, 565.17	72, 013.73	27, 747, 179.49	1, 839, 155.11
200KB – 500KB	1, 735, 774.40	52, 267, 914.50	77, 263.62	68, 936, 117.54	2, 022, 582.43
500KB – 800KB	1, 659, 032.58	82, 978, 099.83	75, 640.55	145, 372, 875.17	1, 986, 880.98
800KB – 1000KB	1, 598, 017.40	88, 494, 890.94	73, 061.13	173, 441, 760.65	1, 943, 260.38
1MB – 5MB	1, 051, 282.38	62, 686, 895.54	75, 588.33	104, 410, 319.78	1, 457, 331.50
5MB – 10MB	1, 507, 586.89	123, 647, 599.33	77, 281.59	235, 930, 306.54	1, 810, 907.29
10MB – 50MB	1, 426, 220.69	50, 851, 625.58	77, 538.46	185, 937, 224.32	1, 871, 615.48
50MB – 60MB	1, 122, 488.77	41, 517, 033.83	71, 350.04	141, 475, 097.98	1, 791, 240.50
60MB – 70MB	1, 149, 169.33	42, 836, 577.30	71, 306.22	121, 760, 402.13	1, 796, 145.29
70MB – 80MB	1, 147, 514.37	8, 779, 142.75	82, 730.19	8, 947, 352.67	1, 802, 628.72
80MB – 100MB	1, 171, 793.67	8, 252, 345.26	79, 880.46	7, 401, 233.92	1, 655, 950.85
100MB ขึ้นไป	1, 035, 611.94	24, 574, 852.03	74, 280.13	118, 221, 969.35	1, 774, 658.91
หมวดหมู่ไฟล์: Group_Music					
1MB – 5MB	3, 614, 358.45	60, 538, 316.04	78, 328.40	358, 438, 109.68	2, 939, 106.19
5MB – 10MB	3, 576, 618.84	152, 783, 179.94	70, 153.17	443, 127, 737.17	2, 997, 992.49
10MB – 50MB	3, 658, 453.31	161, 126, 161.34	72, 801.74	689, 361, 983.66	3, 046, 857.38
50MB – 60MB	3, 596, 765.23	95, 951, 538.97	75, 767.71	916, 899, 571.09	2, 970, 084.19
60MB – 70MB	3, 430, 975.75	107, 460, 389.58	70, 504.08	870, 716, 415.77	3, 063, 803.74
70MB – 80MB	3, 473, 580.00	240, 080, 569.35	72, 340.77	953, 435, 721.66	3, 065, 733.49
80MB – 100MB	3, 405, 849.59	309, 338, 826.85	241, 163.07	955, 076, 565.97	3, 093, 405.63
100MB ขึ้นไป	3, 542, 476.80	183, 978, 755.74	645, 259.33	908, 887, 067.85	3, 006, 343.97
ค่าเฉลี่ยรวม	2, 432, 508.32	97, 018, 493.26	108, 239.54	366, 459, 237.54	2, 318, 286.13

จากผลการทดสอบในตารางที่ 4.4 แสดงอัตราความเร็วในการคลายข้อมูล โดยทำการแบ่งแยกตามช่วงขนาดไฟล์ 15 ช่วงไฟล์ พบว่าไฟล์ขนาด 1 กิโลไบต์ – 10 กิโลไบต์ อัลกอริทึม DEFLATE สามารถทำความเร็วในการคลายข้อมูลได้ดีในหมวดไฟล์รูปภาพและหมวดไฟล์เอกสาร และอัลกอริทึม DEFLATE ยังสามารถทำความเร็วได้ดีในช่วงขนาดไฟล์ 80 เมกกะไบต์ – 100 เมกกะไบต์ ในหมวดไฟล์รูปภาพ แต่ค่าเฉลี่ยรวมกลับพบว่าอัลกอริทึม LZ4 มีอัตราความเร็วในการคลายข้อมูลที่สูงกว่าอัลกอริทึมอื่น ๆ

4.3 ผลการเปรียบเทียบอัตราส่วนการบีบอัดข้อมูล (Compression Ratio)

เป็นการวัดหาอัตราส่วนพื้นที่ขนาดไฟล์คงเหลือหลังจากผ่านขั้นตอนการบีบอัดไฟล์ข้อมูลมาแล้ว ซึ่งผลลัพธ์การเปรียบเทียบอัตราส่วนการบีบอัดไฟล์ข้อมูลที่ได้จะมีหน่วยเป็นร้อยละของขนาดไฟล์ข้อมูลดั้งเดิม โดยสามารถสรุปผลการทดสอบได้ดังตารางที่ 4.5

ตารางที่ 4.5 ร้อยละอัตราส่วนการบีบอัดข้อมูล

อัลกอริทึม	File Group (%)			
	Data	Images	Music & Videos	ค่าเฉลี่ยรวม
BZip2	76.47	76.51	94.82	81.62
DEFLATE	76.21	77.20	96.81	82.30
Huffman Coding	111.43	109.10	98.65	107.11
LZ4	80.01	85.99	98.77	87.18
LZMA	73.05	71.03	94.72	78.52

จากตารางที่ 4.5 แสดงข้อมูลร้อยละอัตราส่วนการบีบอัดข้อมูล ผลการทดลองเปรียบเทียบอัตราส่วนการบีบอัดข้อมูลระหว่างอัลกอริทึม LZMA, BZip2, DEFLATE, LZ4, Huffman Coding พบว่าอัลกอริทึม LZMA มีอัตราส่วนการบีบอัดที่น้อยที่สุดใน 3 หมวดไฟล์ โดยเฉลี่ย 78.52% อัลกอริทึม BZip2 โดยเฉลี่ย 81.62% อัลกอริทึม DEFLATE โดยเฉลี่ย 82.30% อัลกอริทึม LZ4 โดยเฉลี่ย 87.18% และอัลกอริทึม Huffman Coding โดยเฉลี่ย 107.11% และพบว่าหมวดไฟล์ที่มีอัตราส่วนการบีบอัดที่น้อยที่สุด คือ หมวดไฟล์รูปภาพ โดยมีค่าเฉลี่ย 71.03% ซึ่งหมายความว่าเมื่อนำไฟล์เอกสารใด ๆ มาผ่านกระบวนการบีบอัดข้อมูลจะทำให้มีขนาดที่เล็กลง ยังมีอัตราส่วนการบีบอัดที่น้อยก็จะทำให้ขนาดไฟล์เล็กลงไปเรื่อย ๆ ดังนั้นหากพิจารณาในด้านการลดขนาดของไฟล์ อัลกอริทึม LZMA จึงมีประสิทธิภาพดีที่สุด จากผลการทดลองยังพบว่าอัลกอริทึม Huffman Coding มีอัตราส่วนการบีบอัดที่มีค่ามากที่สุด โดยค่าเฉลี่ย 107.11% ทั้งในหมวดไฟล์ข้อมูล หมวดไฟล์

รูปภาพ และหมวดไฟล์เพลงและวิดีโอ อัลกอริทึม Huffman Coding เมื่อไฟล์มีขนาดใหญ่ ถ้าทำการบีบอัดไฟล์จะพบว่าขนาดไฟล์หลังจากบีบอัดจะมีขนาดที่ใหญ่ขึ้นกว่าไฟล์ต้นฉบับ หรือขนาดไฟล์ลดลงแต่ไม่มากนัก เพราะฉะนั้นอัลกอริทึม Huffman Coding จึงไม่ควรนำมาบีบอัดข้อมูลที่เน้นลดขนาดไฟล์

ตารางที่ 4.6 ร้อยละอัตราส่วนการบีบอัดข้อมูลแยกตามช่วงขนาดไฟล์ข้อมูล

ช่วงขนาดไฟล์	File Group (%)				
	BZip2	DEFLATE	Huffman Coding	LZ4	LZMA
หมวดหมู่ไฟล์: Group_Data					
1KB – 5KB	50.59	39.52	453.39	56.44	41.90
5KB – 10KB	53.59	48.99	109.82	58.91	49.17
10KB – 50KB	52.47	48.76	81.05	51.64	47.23
50KB – 200KB	80.84	79.27	93.63	82.25	78.51
200KB – 500KB	57.93	57.75	87.06	62.84	54.44
500KB – 800KB	93.79	93.23	100.32	94.68	92.80
800KB – 1000KB	63.65	63.56	88.69	65.70	62.20
1MB – 5MB	74.23	74.99	88.12	77.47	72.72
5MB – 10MB	71.54	72.24	87.07	75.33	63.28
10MB – 50MB	82.98	84.62	96.02	87.94	81.54
50MB – 60MB	98.46	98.34	99.97	98.72	98.42
60MB – 70MB	93.73	95.48	99.99	98.95	91.82
70MB – 80MB	100.27	99.91	100.02	99.95	96.02
80MB – 100MB	92.40	95.44	99.98	99.09	90.07
100MB ขึ้นไป	92.21	95.31	99.97	99.07	89.77
หมวดหมู่ไฟล์: Group_Images					
1KB – 5KB	154.04	82.19	305.70	82.61	78.28
5KB – 10KB	109.11	99.44	139.29	99.94	98.90
10KB – 50KB	54.17	50.61	76.32	54.35	48.60
50KB – 200KB	78.67	76.22	101.10	78.35	74.94
200KB – 500KB	98.06	97.59	100.96	98.39	96.61
500KB – 800KB	98.96	98.30	100.14	99.13	97.33
800KB – 1000KB	87.21	90.61	97.87	95.72	85.78
1MB – 5MB	46.53	53.02	83.00	62.46	44.00
5MB – 10MB	80.22	87.17	93.81	97.53	80.01
10MB – 50MB	72.26	81.71	97.65	93.74	74.60

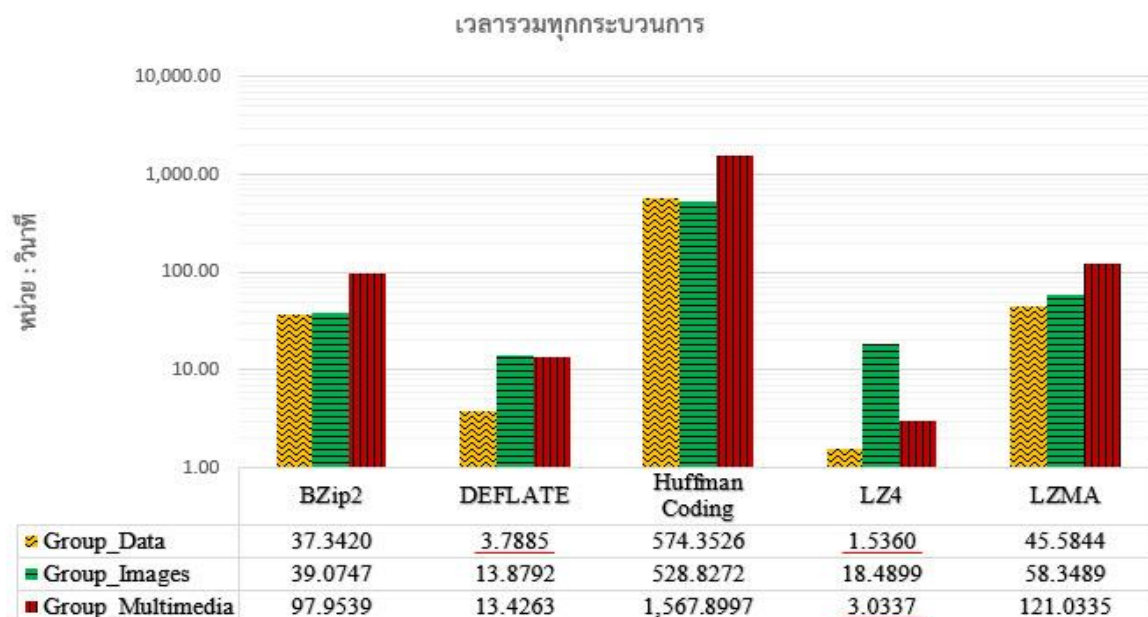
ตารางที่ 4.6 ร้อยละอัตราส่วนการบีบอัดข้อมูลแยกตามช่วงขนาดไฟล์ข้อมูล

ช่วงขนาดไฟล์	File Group (%)				
	BZip2	DEFLATE	Huffman Coding	LZ4	LZMA
50MB – 60MB	50.67	69.04	96.92	92.76	53.97
60MB – 70MB	51.83	70.21	96.69	92.72	55.06
70MB – 80MB	52.63	69.95	96.02	92.69	64.75
80MB – 100MB	40.03	52.59	75.34	65.88	39.97
100MB ขึ้นไป	44.94	64.81	96.61	90.74	47.53
หมวดหมู่ไฟล์: Group_Music					
1MB – 5MB	96.87	96.88	99.46	97.39	97.04
5MB – 10MB	95.94	95.99	98.96	96.84	96.07
10MB – 50MB	96.49	96.78	99.12	97.96	96.41
50MB – 60MB	96.74	97.68	98.58	99.62	96.65
60MB – 70MB	91.57	96.25	98.31	99.47	91.16
70MB – 80MB	92.53	97.02	98.43	99.85	92.33
80MB – 100MB	92.34	96.61	98.23	99.57	91.83
100MB ขึ้นไป	95.38	97.14	98.07	99.44	95.58
ค่าเฉลี่ยรวม	81.62	82.30	107.11	87.18	78.52

จากผลการทดสอบในตารางที่ 4.6 แสดงอัตราส่วนการบีบอัดข้อมูลโดยทำการแบ่งแยกตามช่วงขนาดไฟล์ 15 ช่วงไฟล์ พบว่าไฟล์ขนาด 10 เมกะไบต์ขึ้นไป อัลกอริทึม BZip2 มีอัตราการบีบอัดข้อมูลน้อยกว่าอัลกอริทึมอื่น ๆ ในหมวดไฟล์รูปภาพ และช่วงขนาดไฟล์ 1 เมกะไบต์ – 10 เมกะไบต์ ในหมวดไฟล์เพลงและวิดีโอ จึงทำให้ไฟล์ที่ได้มีขนาดเล็กกว่าอัลกอริทึมอื่น ๆ แต่กลับพบว่าค่าเฉลี่ยรวมที่มีอัตราส่วนการบีบอัดดีที่สุด เป็นอัลกอริทึม LZMA ที่มีค่าเฉลี่ยรวม 78.52%

4.4 ผลการเปรียบเทียบประสิทธิภาพเวลารวมทุกกระบวนการ (Overall Time)

การวัดอัตราเวลารวมทุกกระบวนการทำงาน (Overall Time) โดยจะวัดประสิทธิภาพการทำงานตั้งแต่เริ่มต้นการบีบอัดข้อมูล การรับ-ส่งข้อมูลส่งผ่านเว็บแอปพลิเคชัน จนถึงขั้นตอนการคลายข้อมูล สามารถแสดงผลทดลองได้ดังภาพที่ 4.1



ภาพที่ 4.1 อัตราความเร็วทุกกระบวนการตั้งแต่เริ่มบีบอัดข้อมูลถึงกระบวนการคลายข้อมูล

จากผลการทดลองในภาพที่ 4.1 พบว่า อัลกอริทึม LZ4 มีค่าเฉลี่ย Overall Time น้อยที่สุด โดยมีค่าเฉลี่ย 7.3806 วินาที และพบว่ามีค่าเฉลี่ยที่ต่ำในหมวดไฟล์ข้อมูล โดยเฉลี่ย 1.5360 วินาที และหมวดไฟล์มัลติมีเดีย โดยเฉลี่ย 3.0337 วินาที แต่พบว่าในหมวดไฟล์รูปภาพ อัลกอริทึมที่มีค่าเฉลี่ยที่น้อยที่สุดคือ อัลกอริทึม Deflate โดยเฉลี่ย 13.8792 วินาที เมื่อดูค่า Overall Time โดยเฉลี่ยรวมทั้งหมดกลับพบว่า อัลกอริทึม LZ4 มีค่าเฉลี่ย Overall Time น้อยที่สุด ทำให้การส่งข้อมูลผ่านเว็บแอปพลิเคชันมีประสิทธิภาพและมีความเร็วเพิ่มขึ้น จากตารางที่ 4.5 จะพบว่าอัลกอริทึม LZMA จะมีอัตราส่วนการบีบอัดที่ดีที่สุด แต่เมื่อเทียบประสิทธิภาพรวมทุกกระบวนการ อัลกอริทึม LZMA ยังเป็นรองอัลกอริทึม LZ4 อยู่มาก จากรูปที่ 4.1 จะพบว่าอัตราส่วนเฉลี่ยค่า Overall Time อัลกอริทึม LZ4 จะมีค่าเฉลี่ยน้อยกว่า อัลกอริทึม LZMA โดยเฉลี่ย 63.4977 วินาที

พหุ อนุ ทั โตะ ชีเว

ตารางที่ 4.7 อัตราความเร็วรวมทุกกระบวนการ แยกตามช่วงขนาดไฟล์ข้อมูล

ช่วงขนาดไฟล์	File Group (byte / s)				
	BZip2	DEFLATE	Huffman Coding	LZ4	LZMA
หมวดหมู่ไฟล์: Group_Data					
1KB – 5KB	0.30	0.03	0.08	0.08	0.05
5KB – 10KB	0.06	0.05	0.17	0.05	0.08
10KB – 50KB	0.07	0.07	0.41	0.07	0.15
50KB – 200KB	0.19	0.09	2.17	0.42	0.23
200KB – 500KB	0.41	0.17	6.77	0.15	0.53
500KB – 800KB	1.17	1.30	17.02	1.37	1.26
800KB – 1000KB	1.18	0.28	19.92	0.29	1.33
1MB – 5MB	3.36	1.66	54.60	0.24	3.90
5MB – 10MB	8.81	2.38	179.20	0.74	10.55
10MB – 50MB	31.11	6.99	482.35	1.12	39.53
50MB – 60MB	104.58	5.48	1, 529.42	2.85	122.85
60MB – 70MB	113.10	6.97	1, 892.34	4.19	143.16
70MB – 80MB	138.65	6.46	2, 051.69	3.37	147.26
80MB – 100MB	158.86	10.03	2, 478.44	6.08	199.06
100MB ขึ้นไป	331.54	23.97	4, 867.06	14.15	418.77
หมวดหมู่ไฟล์: Group_Images					
1KB – 5KB	0.35	0.05	0.07	0.05	0.11
5KB – 10KB	0.10	0.07	0.25	0.17	0.12
10KB – 50KB	0.32	0.07	0.62	0.06	0.15
50KB – 200KB	0.34	0.13	3.40	0.11	0.32
200KB – 500KB	0.83	0.45	7.12	0.18	0.80
500KB – 800KB	1.83	0.36	18.00	0.51	1.98
800KB – 1000KB	2.45	0.92	27.17	0.49	2.73
1MB – 5MB	3.35	0.56	49.49	0.62	4.04
5MB – 10MB	12.60	3.07	141.85	2.78	16.76
10MB – 50MB	59.37	36.78	763.35	36.96	89.02
50MB – 60MB	107.81	38.91	1, 701.14	52.36	241.66
60MB – 70MB	126.63	37.12	1, 925.76	45.02	208.59
70MB – 80MB	205.98	43.29	1, 797.97	55.21	219.98
80MB – 100MB	131.59	39.62	1, 946.54	66.53	213.37
100MB ขึ้นไป	217.10	104.10	3, 489.26	156.17	389.59
หมวดหมู่ไฟล์: Group_Music					
1MB – 5MB	7.04	3.32	117.02	0.59	7.99
5MB – 10MB	10.26	4.85	262.65	0.33	11.95
10MB – 50MB	55.56	20.92	1, 006.15	2.06	66.49

ตารางที่ 4.7 อัตราความเร็วรวมทุกกระบวนการ แยกตามช่วงขนาดไฟล์ข้อมูล

ช่วงขนาดไฟล์	File Group (byte / s)				
	BZip2	DEFLATE	Huffman Coding	LZ4	LZMA
50MB – 60MB	89.44	16.71	1, 523.94	2.39	108.96
60MB – 70MB	103.06	7.51	1, 935.51	3.22	129.19
70MB – 80MB	117.37	21.95	2, 388.75	3.75	145.92
80MB – 100MB	145.34	9.88	2, 722.97	3.94	179.19
100MB ขึ้นไป	265.75	20.11	2, 707.36	8.31	331.89
ค่าเฉลี่ยรวม	54.87	9.72	837.98	7.38	70.88

จากผลการทดสอบในตารางที่ 4.7 แสดงอัตราความเร็วรวมทุกกระบวนการ โดยทำการแบ่งแยกตามช่วงขนาดไฟล์ 15 ช่วงไฟล์ พบว่าไฟล์ขนาด 5 กิโลไบต์ – 100 เมกกะไบต์ขึ้นไปในหมวดไฟล์รูปภาพ อัลกอริทึม **DEFLATE** สามารถทำความเร็วรวมได้ดีที่สุด แต่ค่าเฉลี่ยรวมกลับพบว่าอัลกอริทึม LZ4 มีอัตราความเร็วรวมทุกกระบวนการทำงานที่สูงกว่าอัลกอริทึมอื่น ๆ โดยมีค่าเฉลี่ยรวมที่ 7.38 ไบต์ต่อวินาที และ อัลกอริทึม DEFLATE ที่มีลำดับถัดมาที่ ค่าเฉลี่ยรวมที่ 9.72 ไบต์ต่อวินาที

จากผลการทดลองอัตราความเร็วรวมทุกกระบวนการทำงาน โดยทำการวัดประสิทธิภาพตั้งแต่เริ่มต้นการบีบอัด การรับ-ส่งข้อมูล จนถึงขั้นตอนในการคลายข้อมูล จากตารางที่ 4.6 และตารางที่ 4.7 เป็นการหาค่าเฉลี่ยรวมทั้งกระบวนการ โดยทำการรวมเวลาในการรับ-ส่งข้อมูลเข้าไปด้วย ดังนั้นผู้วิจัย จึงได้ทำการแยกรายงานผลการทดลอง อัตราความเร็วในการรับ-ส่งข้อมูล โดยจัดแยกตามประเภทไฟล์ ดังตารางที่ 4.8

ตารางที่ 4.8 อัตราความเร็วในการรับ-ส่งข้อมูล แยกตามช่วงขนาดไฟล์ข้อมูล

ช่วงขนาดไฟล์	ค่าเฉลี่ยรวม (byte / s)
1KB – 5KB	0.10
5KB – 10KB	0.08
10KB – 50KB	0.13
50KB – 200KB	0.39
200KB – 500KB	0.90
500KB – 800KB	2.30
800KB – 1000KB	2.95
1MB – 5MB	7.52

ตารางที่ 4.8 อัตราความเร็วในการรับ-ส่งข้อมูล แยกตามช่วงขนาดไฟล์ข้อมูล

ช่วงขนาดไฟล์	ค่าเฉลี่ยรวม (byte / s)
5MB – 10MB	20.52
10MB – 50MB	77.02
50MB – 60MB	170.29
60MB – 70MB	205.77
70MB – 80MB	255.60
80MB – 100MB	303.99
100MB ขึ้นไป	365.49

จากผลการทดลองในตารางที่ 4.8 แสดงอัตราความเร็วในการรับ-ส่งไฟล์ข้อมูลผ่านเว็บแอปพลิเคชันที่มีการบีบอัดข้อมูลมาก่อนแล้ว จึงทำให้อัตราความเร็วในการรับ-ส่งมีประสิทธิภาพมากยิ่งขึ้น จากข้อมูลในตารางที่ 4.8 จะจัดแบ่งข้อมูลออกตามช่วงขนาดของไฟล์ ซึ่งจากผลการทดลองจะพบว่าอัตราความเร็วเฉลี่ยจะมีเวลาที่ไล่เรียงตามขนาดของไฟล์ เมื่อขนาดไฟล์ที่ต่ำกว่า 5 เมกะไบต์ อัตราความเร็วเฉลี่ยอยู่ในช่วงไม่เกิน 10 ไบต์ต่อวินาที ขนาดไฟล์ตั้งแต่ 5 เมกะไบต์ขึ้นไป อัตราความเร็วเฉลี่ยอยู่ที่ประมาณ 70-365 ไบต์ต่อวินาที

4.5 ผลการทดสอบประสิทธิภาพการบีบอัด/การคลายข้อมูล ตามช่วงกลุ่มขนาดไฟล์

เป็นการออกแบบกลุ่มช่วงขนาดไฟล์ข้อมูล เพื่อทำการทดสอบประสิทธิภาพความเร็วในการบีบอัด ความเร็วในการคลายข้อมูล อัตราส่วนการบีบอัด และเวลารวมทุกกระบวนการ โดยการจัดกลุ่มข้อมูลใหม่ จากเดิมที่มีขนาดไฟล์ข้อมูล จำนวน 15 ขนาดไฟล์ โดยขนาดไฟล์นี้ เป็นการจัดลำดับขนาดไฟล์ที่เริ่มต้นจากขนาดเล็กไปจนถึงขนาดใหญ่ และมีขนาดที่ใกล้เคียงกันมาก ดังนั้นผู้วิจัยได้ทำการจัดกลุ่มช่วงขนาดไฟล์ใหม่ โดยการจัดกลุ่มข้อมูลใหม่ออกเป็น 3 ช่วงข้อมูล คือ 1.กลุ่มไฟล์ขนาดเล็ก (1 กิโลไบต์ – 10 เมกะไบต์) 2.กลุ่มไฟล์ขนาดกลาง (10 เมกะไบต์– 70 เมกะไบต์) 3.กลุ่มไฟล์ขนาดใหญ่ (70 เมกะไบต์ ขึ้นไป) จากผลการทดสอบประสิทธิภาพในแต่ละช่วงกลุ่มข้อมูล สามารถแสดงผลการทดลองได้ ดังตารางที่ 4.9

ตารางที่ 4.9 ผลการเปรียบเทียบประสิทธิภาพ ตามช่วงกลุ่มขนาดไฟล์

อัลกอริทึม	Compression	Decompression	Overall	Compression
	Time (byte / s)	Time (byte / s)	Time (byte / s)	Ratio (%)
กลุ่มไฟล์ขนาดเล็ก	16, 947, 076.75	41, 517, 963.64	15.82	84.89
LZMA	870, 626.30	1, 970, 743.29	4.80	73.33
BZip2	1, 521, 146.10	2, 033, 930.90	4.01	79.11
Huffman Coding	2, 130, 560.46	89, 157.37	68.35	114.74
DEFLATE	27, 439, 767.91	55, 885, 437.66	1.39	76.55
LZ4	52, 773, 282.98	147, 610, 549.00	0.57	80.70
กลุ่มไฟล์ขนาดกลาง	25, 354, 073.33	144, 150, 499.95	284.11	90.94
LZMA	758, 765.42	2, 735, 691.08	100.42	85.90
BZip2	1, 171, 837.72	2, 965, 866.05	73.93	85.94
Huffman Coding	2, 417, 174.49	76, 612.68	1, 220.47	97.90
DEFLATE	23, 993, 500.72	119, 711, 042.55	16.05	89.84
LZ4	98, 429, 088.29	595, 263, 287.39	9.65	95.09
กลุ่มไฟล์ขนาดใหญ่	30, 542, 785.82	184, 338, 091.13	634.37	89.89
LZMA	748, 760.36	2, 815, 009.41	240.04	84.24
BZip2	1, 244, 488.62	2, 902, 121.34	185.54	83.19
Huffman Coding	2, 457, 197.87	213, 289.52	2, 679.90	96.65
DEFLATE	24, 284, 113.79	192, 209, 872.82	26.47	89.25
LZ4	122, 875, 886.05	716, 827, 085.43	25.30	95.89

จากข้อมูลในตารางที่ 4.9 แสดงผลการเปรียบเทียบประสิทธิภาพ ความเร็วในการบีบอัด ความเร็วในการคลายข้อมูล อัตราส่วนการบีบอัด และเวลารวมทุกกระบวนการ โดยทำการจัดแยกข้อมูลตามช่วงขนาดไฟล์ อัลกอริทึม LZ4 สามารถทำความเร็วในการบีบอัด/การคลายข้อมูล และเวลารวมทุกกระบวนการ ได้ดีที่สุดในทุกช่วงกลุ่มขนาดไฟล์ข้อมูล และผลการเปรียบเทียบด้านอัตราส่วนการบีบอัดข้อมูล พบว่าอัลกอริทึม LZMA มีอัตราส่วนการบีบอัดที่มีประสิทธิภาพมากที่สุด ในช่วงกลุ่มไฟล์ขนาดเล็กและขนาดกลาง แต่กลับพบว่าอัลกอริทึม BZip2 มีอัตราส่วนการบีบอัดดีที่สุดในช่วงกลุ่มไฟล์ขนาดใหญ่

4.6 ผลการทดสอบประสิทธิภาพการบีบอัดด้วยสถิติเชิงอนุมาน

เป็นการวัดผลการทดสอบประสิทธิภาพการบีบอัดข้อมูล โดยใช้หลักการทางสถิติเชิงอนุมานเข้ามาทดสอบ เพื่อให้ได้การทดสอบที่มีประสิทธิภาพด้านความเร็วดีที่สุดไปใช้เป็นต้นแบบการบีบอัด จึง

ได้นำหลักการสถิติเชิงอนุมานเข้ามาเปรียบเทียบผลการบีบอัด โดยการวิเคราะห์สมมติฐาน และทำการทดสอบหาความแตกต่างค่าที (t-test) ในกรณีการเปรียบเทียบประสิทธิภาพอัลกอริทึม ทั้ง 5 อัลกอริทึม ด้วยการศึกษาค่าความแตกต่างของอัตราความเร็วรวมทุกกระบวนการของแต่ละอัลกอริทึม โดยใช้อัลกอริทึม LZ4, BZip2, DEFLATE, Huffman Coding, LZMA มาเป็นข้อมูลตั้งต้น จากนั้นนำไปเปรียบเทียบหาความแตกต่างกับอัลกอริทึมอื่น ๆ โดยจะทำการทดสอบทีละคู่ไปเรื่อย ๆ จนครบทุกคู่ เพื่อหาว่าแต่ละอัลกอริทึมมีประสิทธิภาพด้านความเร็วดีกว่าอัลกอริทึมอื่น ๆ หรือไม่ จากการทดสอบสามารถแสดงรายละเอียดได้ดังนี้

ตารางที่ 4.10 ผลการทดสอบหาความแตกต่างค่าที (t-test) ตามหมวดไฟล์ข้อมูล

หมวดไฟล์ข้อมูล จัดกลุ่มตามช่วงขนาดไฟล์					Overall Time (Byte / s)
ช่วงไฟล์ ขนาดเล็ก	LZ4 (Avg: 0.40) (SD: 0.30)	BZip2 (Avg: 3.55) (SD: 17.59)	DEFLATE (Avg: 1.21) (SD: 3.34)	Huffman Coding (Avg: 67.20) (SD: 7,391.94)	LZMA (Avg: 4.19) (SD: 25.68)
LZ4		4.76/0.00	2.92/0.00	4.66/0.00	4.62/0.00
BZip2	-4.76/0.00		-4.41/0.00	4.59/0.00	2.50/0.01
DEFLATE	-2.92/0.00	4.41/0.00		4.65/0.00	4.32/0.00
Huffman Coding	-4.66/0.00	-4.59/0.00	-4.65/0.00		-4.59/0.00
LZMA	-4.62/0.00	-2.50/0.01	-4.32/0.00	4.59/0.00	
ช่วงไฟล์ ขนาดกลาง	LZ4 (Avg: 1.71) (SD: 1.36)	BZip2 (Avg: 55.02) (SD: 1,514.06)	DEFLATE (Avg: 5.38) (SD: 34.83)	Huffman Coding (Avg: 847.66) (SD: 352,400.36)	LZMA (Avg: 67.13) (SD: 2,073.29)
LZ4		5.26/0.00	2.66/0.00	5.34/0.00	5.50/0.00
BZip2	-5.26/0.00		-4.98/0.00	5.34/0.00	5.10/0.00
DEFLATE	-2.66/0.00	4.98/0.00		5.33/0.00	5.32/0.00
Huffman Coding	-5.34/0.00	-5.34/0.00	-5.33/0.00		-5.33/0.00
LZMA	-5.50/0.00	-5.10/0.00	-5.32/0.00	5.33/0.00	
ช่วงไฟล์	LZ4	BZip2	DEFLATE	Huffman	LZMA

ตารางที่ 4.10 ผลการทดสอบหาความแตกต่างค่าที่ (t-test) ตามหมวดไฟล์ข้อมูล

หมวดไฟล์ข้อมูล จัดกลุ่มตามช่วงขนาดไฟล์				Overall Time (Byte / s)	
ขนาดใหญ่	(Avg: 7.86) (SD: 26.63)	(Avg: 209.67) (SD: 9,266.02)	(Avg: 13.48) (SD: 72.41)	Coding (Avg: 3,132.39) (SD: 1,848,933.56)	(Avg: 255.03) (SD: 16,880.71)
LZ4		5.39/0.00	4.09/0.00	5.64/0.00	4.85/0.00
BZip2	-5.39/0.00		-5.43/0.00	5.66/0.00	2.41/0.03
DEFLATE	-4.09/0.00	5.43/0.00		5.65/0.00	4.87/0.00
Huffman Coding	-5.64/0.00	-5.66/0.00	-5.65/0.00		-5.72/0.00
LZMA	-4.85/0.00	-2.41/0.03	-4.87/0.00	5.72/0.00	

*มีนัยสำคัญทางสถิติที่ระดับ 0.05

คำอธิบายตัวเลข : ค่าที่ (t-test) / P value

จากตารางที่ 4.10 แสดงผลการทดสอบค่า t-test แยกตามหมวดไฟล์ข้อมูลประกอบด้วย ช่วงไฟล์ขนาดเล็ก ช่วงไฟล์ขนาดกลาง และช่วงไฟล์ขนาดใหญ่ โดยการนำอัลกอริทึม LZ4, BZip2, DEFLATE, Huffman Coding และ LZMA มาเป็นอัลกอริทึมในการเปรียบเทียบความมีนัยแตกต่างของอัลกอริทึมอื่น ๆ ซึ่งสามารถจัดกลุ่มการทดสอบได้ดังนี้

1) ความมีนัยแตกต่างของอัลกอริทึม **LZ4** ต่อ (BZip2, DEFLATE, Huffman Coding, LZMA) จากผลการทดสอบพบว่า อัลกอริทึม LZ4 มีอัตราความเร็วดีกว่าอัลกอริทึมตัวอื่น ๆ อย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05 ในทุกช่วงขนาดไฟล์

2) ความมีนัยแตกต่างของอัลกอริทึม **BZip2** ต่อ (LZ4, DEFLATE, Huffman Coding, LZMA) จากผลการทดสอบพบว่า อัลกอริทึม BZip2 มีอัตราความเร็วดีกว่าอัลกอริทึม Huffman Coding และ LZMA ในทุกช่วงขนาดไฟล์ อย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05 แต่มีความเร็วดีน้อยกว่าอัลกอริทึม LZ4 และอัลกอริทึม DEFLATE อย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05 ในทุกช่วงขนาดไฟล์

3) ความมีนัยแตกต่างของอัลกอริทึม **DEFLATE** ต่อ (BZip2, LZ4, Huffman Coding, LZMA) จากผลการทดสอบพบว่า อัลกอริทึม DEFLATE มีอัตราความเร็วดีกว่าอัลกอริทึม BZip2, Huffman Coding และ LZMA ในทุกช่วงขนาดไฟล์ อย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05 แต่มีความเร็วดีน้อยกว่าอัลกอริทึม LZ4 อย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05 ในทุกช่วงขนาดไฟล์

4) ความมีนัยแตกต่างของอัลกอริทึม Huffman Coding ต่อ (BZip2, DEFLATE, LZ4, LZMA) จากผลการทดสอบพบว่า อัลกอริทึม Huffman Coding มีอัตราความเร็วต่ำกว่าอัลกอริทึมตัวอื่น ๆ อย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05 ในทุกช่วงขนาดไฟล์

5) ความมีนัยแตกต่างของอัลกอริทึม LZMA ต่อ (BZip2, DEFLATE, Huffman Coding, LZ4) จากผลการทดสอบพบว่า อัลกอริทึม LZMA มีอัตราความเร็วดีกว่าอัลกอริทึม Huffman Coding ในทุกช่วงขนาดไฟล์ อย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05 แต่มีความเร็วต่ำกว่าอัลกอริทึม LZ4, DEFLATE และ BZip2 อย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05 ในทุกช่วงขนาดไฟล์

ตารางที่ 4.11 ผลการทดสอบหาความแตกต่างค่าที (t-test) หมวดไฟล์รูปภาพ

หมวดไฟล์รูปภาพ จัดกลุ่มตามช่วงขนาดไฟล์					Overall Time (Byte / s)
ช่วงไฟล์ ขนาดเล็ก	LZ4 (Avg: 0.74) (SD: 1.48)	BZip2 (Avg: 3.40) (SD: 19.95)	DEFLATE (Avg: 0.85) (SD: 1.79)	Huffman Coding (Avg: 39.20) (SD: 2,685.34)	LZMA (Avg: 4.25) (SD: 37.54)
LZ4		4.40/0.00	1.00/0.16	4.48/0.00	3.93 /0.00
BZip2	-4.40/0.00		-4.35/0.00	4.46/0.00	2.09/0.02
DEFLATE	-1.00/0.16	4.35/0.00		4.47/0.00	3.82/0.00
Huffman Coding	-4.48/0.00	-4.46/0.00	-4.47/0.00		-4.48/0.00
LZMA	-3.93/0.00	-2.09/0.02	-3.82/0.00	4.48/0.00	
ช่วงไฟล์ ขนาดกลาง	LZ4 (Avg: 41.27) (SD: 395.18)	BZip2 (Avg: 88.40) (SD: 1,323.50)	DEFLATE (Avg: 35.52) (SD: 324.62)	Huffman Coding (Avg: 1,285.70) (SD: 403,886.83)	LZMA (Avg: 150.54) (SD: 6,413.39)
LZ4		4.01 /0.00	-1.29/0.13	4.89/0.00	3.92/0.01
BZip2	-4.01 /0.00		-3.76/0.01	4.89 /0.00	3.15/0.00
DEFLATE	1.29/0.13	3.76 /0.01		4.86 /0.00	3.82 /0.01
Huffman Coding	-4.89/0.00	-4.89/0.00	-4.86/0.00		-4.95/0.00
LZMA	-3.92/0.01	-3.15/0.01	-3.82/0.01	4.95/0.00	
ช่วงไฟล์	LZ4	BZip2	DEFLATE	Huffman	LZMA

ตารางที่ 4.11 ผลการทดสอบหาความแตกต่างค่าที (t-test) หมวดไฟล์รูปภาพ

หมวดไฟล์รูปภาพ จัดกลุ่มตามช่วงขนาดไฟล์				Overall Time (Byte / s)	
ขนาดใหญ่	(Avg: 84.60) (SD: 1,839.67)	(Avg: 178.59) (SD: 2,195.72)	(Avg: 58.59) (SD: 358.26)	Coding (Avg: 2,369.37) (SD: 122,805.74)	(Avg: 261.30) (SD: 3,222.42)
LZ4		9.75/0.01	-0.94/0.22	36.87/0.00	7.61/0.01
BZip2	-9.75/0.01		-3.91/0.03	40.05/0.00	4.85/0.02
DEFLATE	0.94/0.22	3.91/0.03		48.44/0.00	10.42/0.00
Huffman Coding	-36.87/0.00	-40.05/0.00	-48.44/0.00		-54.58/0.00
LZMA	-7.61/0.01	-4.85/0.02	-10.42/0.00	54.58/0.00	

*มีนัยสำคัญทางสถิติที่ระดับ 0.05

คำอธิบายตัวเลข : ค่าที (t-test) / P value

จากตารางที่ 4.11 แสดงผลการทดสอบค่า t-test แยกตามหมวดไฟล์รูปภาพประกอบด้วย ช่วงไฟล์ขนาดเล็ก ช่วงไฟล์ขนาดกลาง และช่วงไฟล์ขนาดใหญ่ โดยการนำอัลกอริทึม LZ4, BZip2, DEFLATE, Huffman Coding และ LZMA มาเป็นอัลกอริทึมในการเปรียบเทียบความมีนัยแตกต่างของอัลกอริทึมอื่น ๆ ซึ่งสามารถจัดกลุ่มการทดสอบได้ดังนี้

1) ความมีนัยแตกต่างของอัลกอริทึม **LZ4** ต่อ (BZip2, DEFLATE, Huffman Coding, LZMA) จากผลการทดสอบพบว่า อัลกอริทึม LZ4 มีอัตราความเร็วดีกว่าอัลกอริทึม BZip2, Huffman Coding, LZMA อย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05 ในทุกช่วงขนาดไฟล์ แต่มีอัตราความเร็วไม่แตกต่างกับอัลกอริทึม DEFLATE อย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05 ในทุกช่วงขนาดไฟล์

2) ความมีนัยแตกต่างของอัลกอริทึม **BZip2** ต่อ (LZ4, DEFLATE, Huffman Coding, LZMA) จากผลการทดสอบพบว่า อัลกอริทึม BZip2 มีอัตราความเร็วดีกว่าอัลกอริทึม Huffman Coding และ LZMA ในทุกช่วงขนาดไฟล์ อย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05 แต่มีความเร็วดีด้อยกว่าอัลกอริทึม LZ4 และอัลกอริทึม DEFLATE อย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05 ในทุกช่วงขนาดไฟล์

3) ความมีนัยแตกต่างของอัลกอริทึม **DEFLATE** ต่อ (BZip2, LZ4, Huffman Coding, LZMA) จากผลการทดสอบพบว่า อัลกอริทึม DEFLATE มีอัตราความเร็วดีกว่าอัลกอริทึม BZip2, Huffman Coding, LZMA อย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05 ในทุกช่วงขนาดไฟล์ แต่มีอัตราความเร็วไม่แตกต่างกับอัลกอริทึม LZ4 อย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05 ในทุกช่วงขนาดไฟล์

4) ความมีนัยแตกต่างของอัลกอริทึม Huffman Coding ต่อ (BZip2, DEFLATE, LZ4, LZMA) จากผลการทดสอบพบว่า อัลกอริทึม Huffman Coding มีอัตราความเร็วต่ำกว่าอัลกอริทึมตัวอื่น ๆ อย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05 ในทุกช่วงขนาดไฟล์

5) ความมีนัยแตกต่างของอัลกอริทึม LZMA ต่อ (BZip2, DEFLATE, Huffman Coding, LZ4) จากผลการทดสอบพบว่า อัลกอริทึม LZMA มีอัตราความเร็วดีกว่าอัลกอริทึม Huffman Coding ในทุกช่วงขนาดไฟล์ อย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05 แต่มีความเร็วต่ำกว่าอัลกอริทึม LZ4, DEFLATE และ BZip2 อย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05 ในทุกช่วงขนาดไฟล์

ตารางที่ 4.12 ผลการทดสอบหาความแตกต่างค่าที (t-test) ตามหมวดไฟล์เสียงและวิดีโอ

หมวดไฟล์เสียงและวิดีโอ จัดกลุ่มตามช่วงขนาดไฟล์					Overall Time (Byte / s)
ช่วงไฟล์ ขนาดเล็ก	LZ4 (Avg: 0.46) (SD: 0.30)	BZip2 (Avg: 8.64) (SD: 5.94)	DEFLATE (Avg: 4.08) (SD: 7.74)	Huffman Coding (Avg: 189.83) (SD: 30,996.67)	LZMA (Avg: 9.97) (SD: 8.17)
LZ4		10.36/0.00	4.06/0.00	3.40/0.00	10.28/0.00
BZip2	-10.36/0.00		-4.36/0.00	3.28/0.00	6.07/0.00
DEFLATE	-4.06/0.00	4.36/0.00		3.38/0.00	5.69/0.00
Huffman Coding	-3.40/0.00	-3.28/0.00	-3.38/0.00		-3.27/0.00
LZMA	-10.28/0.00	-6.07/0.00	-5.69/0.00	3.27/0.00	
ช่วงไฟล์ ขนาดกลาง	LZ4 (Avg: 2.51) (SD: 0.49)	BZip2 (Avg: 81.48) (SD: 555.03)	DEFLATE (Avg: 15.48) (SD: 342.57)	Huffman Coding (Avg: 1,462.23) (SD: 178,201.62)	LZMA (Avg: 99.92) (SD: 836.04)
LZ4		14.07/0.00	2.88/0.01	14.27/0.00	14.13/0.00
BZip2	-14.07/0.00		-8.52/0.00	14.16/0.00	7.19/0.00
DEFLATE	-2.88/0.01	8.52/0.00		14.02/0.00	9.51/0.00
Huffman Coding	-14.27/0.00	-14.16/0.00	-14.02/0.00		-14.13/0.00
LZMA	-14.13/0.00	-7.19/0.00	-9.51/0.00	14.13/0.00	

ตารางที่ 4.12 ผลการทดสอบหาความแตกต่างค่าที่ (t-test) ตามหมวดไฟล์เสียงและวิดีโอ

หมวดไฟล์เสียงและวิดีโอ จัดกลุ่มตามช่วงขนาดไฟล์					Overall Time (Byte / s)
ช่วงไฟล์ ขนาดใหญ่	LZ4 (Avg: 5.42) (SD: 5.78)	BZip2 (Avg: 180.69) (SD: 5,765.32)	DEFLATE (Avg: 17.72) (SD: 386.31)	Huffman Coding (Avg: 2,593.24) (SD: 3,165,598.53)	LZMA (Avg: 221.56) (SD: 10,409.79)
LZ4		8.89/0.00	2.35/0.02	5.44/0.00	8.10/0.00
BZip2	-8.89/0.00		-8.10/0.00	5.14/0.00	4.92/0.00
DEFLATE	-2.35/0.02	8.10/0.00		5.42/0.00	7.60/0.00
Huffman Coding	-5.44/0.00	-5.14/0.00	-5.42/0.00		-5.09/0.00
LZMA	-8.10/0.00	-4.92/0.00	-7.60/0.00	5.09/0.00	

*มีนัยสำคัญทางสถิติที่ระดับ 0.05

คำอธิบายตัวเลข : ค่าที่ (t-test) / P value

จากตารางที่ 4.12 แสดงผลการทดสอบค่า t-test แยกตามหมวดไฟล์เสียงและวิดีโอ ประกอบด้วย ช่วงไฟล์ขนาดเล็ก ช่วงไฟล์ขนาดกลาง และช่วงไฟล์ขนาดใหญ่ โดยการนำอัลกอริทึม LZ4, BZip2, DEFLATE, Huffman Coding และ LZMA มาเป็นอัลกอริทึมในการเปรียบเทียบความ มีนัยแตกต่างของอัลกอริทึมอื่น ๆ ซึ่งสามารถจัดกลุ่มการทดสอบได้ดังนี้

1) ความมีนัยแตกต่างของอัลกอริทึม **LZ4** ต่อ (BZip2, DEFLATE, Huffman Coding, LZMA) จากผลการทดสอบพบว่า อัลกอริทึม LZ4 มีอัตราความเร็วแตกต่างกว่าอัลกอริทึมตัวอื่น ๆ อย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05 ในทุกช่วงขนาดไฟล์

2) ความมีนัยแตกต่างของอัลกอริทึม **BZip2** ต่อ (LZ4, DEFLATE, Huffman Coding, LZMA) จากผลการทดสอบพบว่า อัลกอริทึม BZip2 มีอัตราความเร็วดีกว่าอัลกอริทึม Huffman Coding และ LZMA ในทุกช่วงขนาดไฟล์ อย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05 แต่มีความเร็วต่ำกว่าอัลกอริทึม LZ4 และอัลกอริทึม DEFLATE อย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05 ในทุกช่วงขนาดไฟล์

3) ความมีนัยแตกต่างของอัลกอริทึม **DEFLATE** ต่อ (BZip2, LZ4, Huffman Coding, LZMA) จากผลการทดสอบพบว่า อัลกอริทึม DEFLATE มีอัตราความเร็วดีกว่าอัลกอริทึม BZip2, Huffman Coding และ LZMA ในทุกช่วงขนาดไฟล์ อย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05 แต่มีความเร็วต่ำกว่าอัลกอริทึม LZ4 อย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05 ในทุกช่วงขนาดไฟล์

4) ความมีนัยแตกต่างของอัลกอริทึม **Huffman Coding** ต่อ (BZip2, DEFLATE, LZ4, LZMA) จากผลการทดสอบพบว่า อัลกอริทึม Huffman Coding มีอัตราความเร็วดีด้อยกว่าอัลกอริทึมตัวอื่น ๆ อย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05 ในทุกช่วงขนาดไฟล์

5) ความมีนัยแตกต่างของอัลกอริทึม **LZMA** ต่อ (BZip2, DEFLATE, Huffman Coding, LZ4) จากผลการทดสอบพบว่า อัลกอริทึม LZMA มีอัตราความเร็วดีดต่ำกว่าอัลกอริทึม Huffman Coding ในทุกช่วงขนาดไฟล์ อย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05 แต่มีความเร็วดีดต่ำกว่าอัลกอริทึม LZ4, DEFLATE และ BZip2 อย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05 ในทุกช่วงขนาดไฟล์

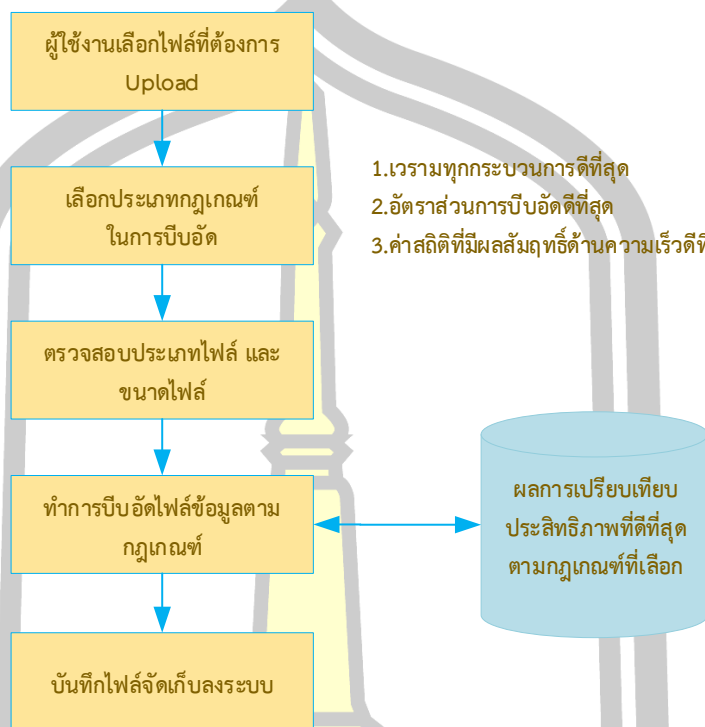
4.7 ผลการทดสอบต้นแบบการบีบอัดข้อมูล

เป็นการออกแบบและพัฒนาต้นแบบการบีบอัดข้อมูล โดยใช้ข้อมูลจากผลการเปรียบเทียบประสิทธิภาพการบีบอัดที่ดีที่สุด 3 ประเภท ประกอบไปด้วย 1) เวลารวมทุกกระบวนการดีที่สุด (ข้อมูลในตารางที่ 4.7) 2) ลดพื้นที่ขนาดไฟล์ดีที่สุด (ข้อมูลในตารางที่ 4.8) 3) ค่าเปรียบเทียบทางสถิติที่มีผลสัมฤทธิ์ด้านความเร็วดีดที่สุด (ข้อมูลในตารางที่ 4.10 - 4.12) โดยทำการแยกข้อมูลตามหมวดหมู่ไฟล์และประเภทไฟล์ข้อมูล จากนั้นนำผลการทดสอบไปสร้างต้นแบบการบีบอัดข้อมูล เพื่อให้การบีบอัดและคลายข้อมูลมีประสิทธิภาพและมีความเหมาะสมกับหมวดหมู่ไฟล์และประเภทไฟล์ต่าง ๆ ในการทดสอบต้นแบบการบีบอัดข้อมูล ผู้วิจัยได้ทำการจัดเตรียมข้อมูลสำหรับการทดสอบการทำงานของระบบ โดยมีรายละเอียดไฟล์ข้อมูลดังตารางที่ 4.13

ตารางที่ 4.13 ข้อมูลสำหรับทดสอบการทำงานของต้นแบบการบีบอัดข้อมูล

ลำดับ ที่	หมวด ไฟล์	ชื่อไฟล์	ขนาดไฟล์	ประเภทการบีบอัด	อัลกอริทึมที่มี ประสิทธิภาพ มากที่สุด
1	Data	Propersal.doc	223 KB	Overall Time ดี ที่สุด	LZ4
2	Images	KTB656.JPG	1.15 MB	Compression Ratio ดีที่สุด	LZMA
3	Music & Videos	ขอโอกาสแทนเด้อ - บอย พนมไพร.mp3	4.86 MB	Overall Time จาก ผลสัมฤทธิ์ทางสถิติ	LZ4
4	Images	ok.tif	132 MB	Overall Time จาก ผลสัมฤทธิ์ทางสถิติ	LZ4

จากข้อมูลในตารางที่ 4.13 เป็นข้อมูลที่ใช้ในการทดสอบการทำงานของต้นแบบการบีบอัดข้อมูล โดยมีขั้นตอนการใช้งานระบบต้นแบบการบีบอัดข้อมูล ได้ดังภาพที่ 4.2



ภาพที่ 4.2 ขั้นตอนในการอัปโหลดไฟล์เข้าระบบและทำการบีบอัดข้อมูล

จากภาพที่ 4.2 เป็นขั้นตอนในการอัปโหลดไฟล์เข้าระบบและทำการบีบอัดไฟล์ข้อมูล ในขั้นตอนนี้ จะทำการเลือกประเภทกฎเกณฑ์ในการบีบอัดก่อน เพื่อใช้สำหรับการเลือกอัลกอริทึมในการบีบอัด ซึ่งผลในการเลือกอัลกอริทึมจะได้จากผลการเปรียบเทียบประสิทธิภาพตามประเภทกฎเกณฑ์ที่เลือกในแต่ละตัว ซึ่งในกฎเกณฑ์จะทำการเปรียบเทียบอัลกอริทึมที่ดีที่สุด และมีความเหมาะสมกับประเภทไฟล์และขนาดไฟล์ เมื่อระบบเลือกอัลกอริทึมที่เหมาะสมกับไฟล์แล้ว ก็จะมีการประมวลผลการบีบอัดและทำการบันทึกและจัดเก็บไฟล์ลงบนพื้นที่จัดเก็บบนเครื่องให้บริการเว็บ จากภาพที่ 4.2 เมื่อนำมาออกแบบและพัฒนาบนเว็บแอปพลิเคชันจะได้ดัง ภาพที่ 4.3

Upload and Auto Compression

อัปโหลดไฟล์จากเครื่องผู้ใช้งานเข้าระบบ เมื่ออัปโหลดไฟล์เสร็จสิ้นระบบจะทำการบีบอัดข้อมูลโดยเลือกวิธีที่ดีที่สุดตามขนาดไฟล์และบันทึกเก็บที่ Server.



File Group :

Data

Images **1**

Music & Videos

กฎเกณฑ์การบีบอัด :

เวลารวมทุกกระบวนการดีที่สุด **2**

ลดพื้นที่ขนาดไฟล์ดีที่สุด

กฎทางสถิติที่มีผลสัมฤทธิ์ด้านความเร็ว ดีที่สุด

Data File :

ok.tif **3**

4

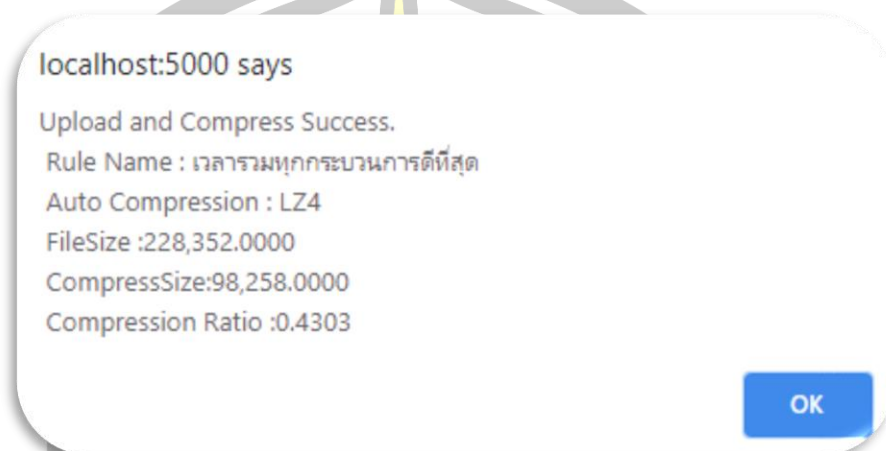
© 2562 - Data Compression By Seksit

ภาพที่ 4.3 ตัวอย่างหน้าจอต้นแบบการบีบอัดข้อมูล

จากภาพที่ 4.3 เป็นรูปภาพหน้าจอของระบบต้นแบบการบีบอัดข้อมูลบนเว็บแอปพลิเคชัน โดยมีขั้นตอนในการทำงานดังนี้ 1) เลือกหมวดหมู่ไฟล์ของไฟล์ที่จะทำการอัปโหลดลงระบบ 2) เลือกกฎเกณฑ์ในการบีบอัดข้อมูล โดยมีกฎเกณฑ์ให้เลือก 3 รูปแบบ ประกอบไปด้วย 1. เวลารวมทุกกระบวนการดีที่สุด (ข้อมูลในตารางที่ 4.7) 2. ลดพื้นที่ขนาดไฟล์ดีที่สุด (ข้อมูลในตารางที่ 4.8) 3. ค่าเปรียบเทียบทางสถิติที่มีผลสัมฤทธิ์ด้านความเร็วดีที่สุด (ข้อมูลในตารางที่ 4.10 - 4.12) 3) เลือกไฟล์ที่ต้องการจะอัปโหลด 4) กดปุ่ม **Submit To Server** เมื่อกดปุ่มโปรแกรมจะทำการอัปโหลดไฟล์จากเครื่องผู้ใช้งานลงจัดเก็บที่เครื่องให้บริการก่อน จากนั้นจะทำการตรวจสอบหมวดหมู่ไฟล์ ประเภทไฟล์ และขนาดไฟล์ จากไฟล์ที่อัปโหลดเข้ามา จากนั้นจะนำข้อมูลที่ได้เข้าไปเปรียบเทียบตามกฎเกณฑ์การบีบอัดที่ผู้ใช้เลือกเข้ามา เมื่อระบบประมวลผลเปรียบเทียบไฟล์เสร็จสิ้น จะได้อัลกอริทึมที่ดีที่สุดและมีความเหมาะสมกับไฟล์ที่สุด ในขั้นตอนต่อไประบบจะนำไฟล์ไปประมวลผลการบีบอัดด้วยอัลกอริทึมที่ได้จากผลการเปรียบเทียบ เมื่อระบบทำการบีบอัดข้อมูลเสร็จสิ้น ไฟล์จะถูกบันทึกเก็บที่เครื่องให้บริการ

จากการทดสอบต้นแบบการบีบอัดตามตัวอย่างข้อมูลในตารางที่ 4.13 สามารถแสดงผลการทดสอบ ได้ดังนี้

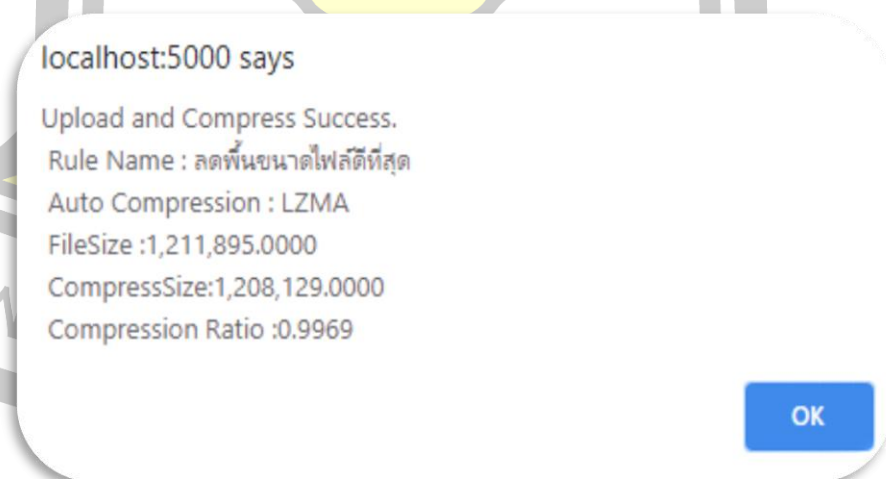
1. หมวดไฟล์ Data ชื่อไฟล์ Propersal.doc ขนาดไฟล์ 223 KB



ภาพที่ 4.4 ตัวอย่างการแจ้งเตือนผลการทดสอบไฟล์ Propersal.doc

ผลการทดสอบไฟล์ Propersal.doc เมื่อนำไฟล์อัปโหลดเข้าระบบเสร็จสิ้น ระบบจะทำการเลือกอัลกอริทึมในการบีบอัดข้อมูลให้เป็น LZ4 และมีอัตราส่วนการบีบอัดที่ 0.4303 เปอร์เซ็นต์

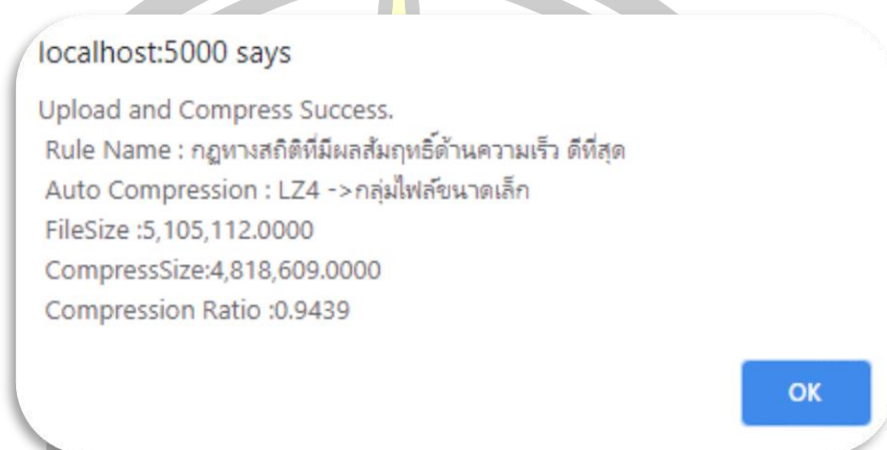
2. หมวดไฟล์ Images ชื่อไฟล์ KTB656.JPG ขนาดไฟล์ 1.15 MB



ภาพที่ 4.5 ตัวอย่างการแจ้งเตือนผลการทดสอบไฟล์ KTB656.JPG

ผลการทดสอบไฟล์ KTB656.JPG เมื่อนำไฟล์อัปโหลดเข้าระบบเสร็จสิ้น ระบบจะทำการเลือกอัลกอริทึมในการบีบอัดข้อมูลให้เป็น LZMA และมีอัตราส่วนการบีบอัดที่ 0.9969 เปอร์เซ็นต์

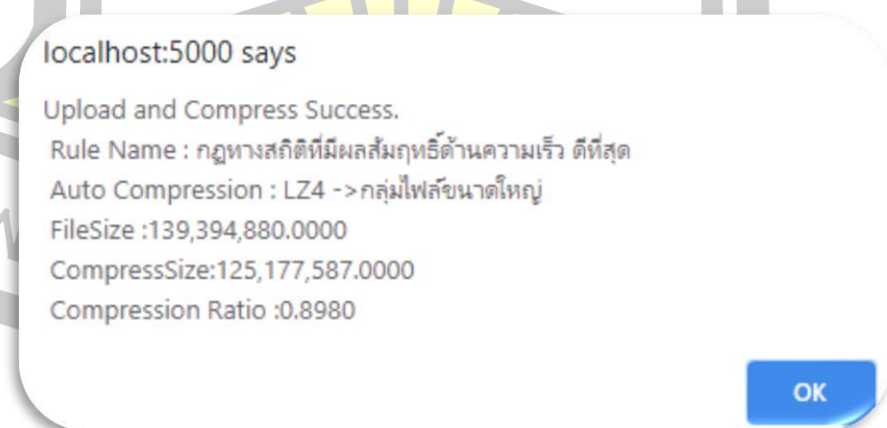
3. หมวดไฟล์ Music & Videos ชื่อไฟล์ ขอโอกาสแหงนเดื่อ-บอยพนมไพร.mp3 ขนาดไฟล์ 4.8MB



ภาพที่ 4.6 ตัวอย่างการแจ้งเตือนผลการทดสอบไฟล์ ขอโอกาสแหงนเดื่อ - บอย พนมไพร.mp3

ผลการทดสอบไฟล์ ขอโอกาสแหงนเดื่อ - บอย พนมไพร.mp3 เมื่อนำไฟล์อัปโหลดเข้าระบบเสร็จสิ้น ระบบจะทำการเลือกอัลกอริทึมในการบีบอัดข้อมูลให้เป็น LZ4 และบอกกลุ่มขนาดไฟล์ที่เลือกใช้ เป็น กลุ่มไฟล์ขนาดเล็ก และมีอัตราส่วนการบีบอัดที่ 0.9439 เปอร์เซ็นต์

4. หมวดไฟล์ Images ชื่อไฟล์ ok.tif ขนาดไฟล์ 132 MB



ภาพที่ 4.7 ตัวอย่างการแจ้งเตือนผลการทดสอบไฟล์ ok.tif

ผลการทดสอบไฟล์ ok.tif เมื่อนำไฟล์อัปโหลดเข้าระบบเสร็จสิ้น ระบบจะทำการเลือกอัลกอริทึมในการบีบอัดข้อมูลให้เป็น LZ4 และบอกกลุ่มขนาดไฟล์ที่เลือกใช้เป็น **กลุ่มไฟล์ขนาดใหญ่** และมีอัตราส่วนการบีบอัดที่ 0.8980 เปอร์เซนต์

จากการทดสอบประสิทธิภาพการบีบอัดข้อมูลบนระบบต้นแบบการบีบอัดข้อมูล พบว่าการบีบอัดไฟล์ข้อมูลจะมีประสิทธิภาพมากยิ่งขึ้น ทั้งในเรื่องของความเร็วในการบีบอัด ความเร็วในการคลายข้อมูล รวมทั้งกระบวนการในการคัดเลือกอัลกอริทึม เนื่องจากระบบต้นแบบได้เลือกอัลกอริทึมที่มีความเหมาะสมกับหมวดหมู่ไฟล์ ประเภทของไฟล์ ให้อัตโนมัติ ผู้ใช้งานไม่จำเป็นต้องเลือกอัลกอริทึมที่จะใช้ในการบีบอัดเอง ทำให้มีความรวดเร็วในการทำงานมากยิ่งขึ้นกว่าเดิม และได้อัลกอริทึมที่มีประสิทธิภาพและเหมาะสมกับหมวดหมู่ไฟล์ และประเภทไฟล์



บทที่ 5

สรุปผลและข้อเสนอแนะ

ผลการวิจัยในการเปรียบเทียบผลการบีบอัดแบบไม่สูญเสียข้อมูลบนเว็บแอปพลิเคชันและนำผลการเปรียบเทียบที่มีประสิทธิภาพมาทำการสร้างต้นแบบการบีบอัดข้อมูล จากการทดลองประสิทธิภาพการทำงานแต่ละอัลกอริทึม สามารถสรุปผลการดำเนินงาน ปัญหาและอุปสรรค และข้อเสนอแนะ ได้ดังต่อไปนี้

5.1 สรุปผลการดำเนินงาน

จากการทดลองเมื่อ พิจารณาจาก Compression Ratio, Compression Time, Decompress Time และ Overall Time พบว่าการใช้การบีบอัดข้อมูลโดยใช้อัลกอริทึมแบบ LZ4 ได้ผลลัพธ์ที่ดีที่สุด ในด้านอัตราความเร็วในการบีบอัด อัตราความเร็วในการคลายข้อมูล และความเร็วรวมทุกกระบวนการ ที่มีอัตราความเร็วที่ดี ทั้งในหมวดไฟล์เอกสาร ไฟล์รูปภาพ และไฟล์เสียงและวิดีโอ จึงเหมาะสำหรับการนำมาใช้ในการบีบอัดข้อมูลและรับ-ส่งข้อมูลผ่านเว็บแอปพลิเคชัน และอัลกอริทึม Huffman Coding ไม่ควรนำมาพิจารณาในการบีบอัดข้อมูล เพราะจากผลการทดลองอัลกอริทึม Huffman Coding จะมีประสิทธิภาพด้อยกว่าอัลกอริทึมอื่น ๆ อย่างมาก ถ้าหากพิจารณาอัลกอริทึมบีบอัดข้อมูลที่เน้นเรื่องการลดขนาดพื้นที่การจัดเก็บไฟล์ พื้นที่ในการสำรองไฟล์ อัลกอริทึม LZMA จะเหมาะสมกับวิธีการนี้มากที่สุด

5.2 อภิปรายผล

งานวิจัยนี้มีวัตถุประสงค์เพื่อเปรียบเทียบอัลกอริทึมการบีบอัดแบบไม่สูญเสียข้อมูลเพื่อลดขนาดไฟล์ข้อมูลให้เล็กลงก่อนทำการส่งไฟล์ผ่านเครือข่ายอินเทอร์เน็ต โดยการเปรียบเทียบผลการบีบอัดและพัฒนาต้นแบบการบีบอัดข้อมูลอัตโนมัติ โดยใช้กฎเกณฑ์การบีบอัดจากผลสรุปการเปรียบเทียบประสิทธิภาพเวลารวมทุกกระบวนการทำงาน

ข้อมูลที่ใช้ในการวิจัยนี้ เป็นข้อมูลไฟล์ข้อมูลทั่วไป โดยทำการจัดแบ่งออกเป็นหมวดไฟล์จำนวน 3 กลุ่ม คือ กลุ่มไฟล์เอกสาร กลุ่มไฟล์รูปภาพ และกลุ่มไฟล์เสียงและวิดีโอ จำนวน 150 ไฟล์ และนำไฟล์ทั้งหมดมาทำการบีบอัดข้อมูลผ่านเว็บแอปพลิเคชันด้วยอัลกอริทึมการบีบอัดแบบไม่สูญเสียข้อมูลจำนวน 5 อัลกอริทึม ได้แก่ Huffman Coding, Deflate, BZip2, LZMA และ LZ4 ซึ่งการนำไฟล์ข้อมูลเข้ามาทดลองในงานวิจัยนี้ อาจได้ค่าผลการทดลองที่แตกต่างกัน ขึ้นอยู่กับขนาดไฟล์ประเภทไฟล์ อัลกอริทึมที่นำมาทดลอง และเงื่อนไขด้านทรัพยากรเครื่องที่ทดสอบ ณ ช่วงเวลาที่แตกต่างกัน ดังนั้นการทดลองการบีบอัด การคลายข้อมูล การรับ-ส่งข้อมูลผ่านเว็บแอปพลิเคชัน อาจจะทำให้ผลการทดลองได้ผลที่แตกต่างกันไป

5.3 ปัญหาและอุปสรรค

จากการดำเนินงานวิจัยผู้วิจัยพบปัญหาและอุปสรรค ซึ่งส่งผลกระทบต่อกระบวนการวิจัย โดยสามารถสรุปประเด็น ได้ดังนี้

5.3.1 การทดสอบประสิทธิภาพการบีบอัดและคลายข้อมูล ต้องทดสอบบนเครื่องบริการเว็บที่มีสถานะการทำงานไม่หนักมากนัก หากเครื่องบริการเว็บทำงานหลาย ๆ งานพร้อม ๆ กัน ส่งผลให้ประสิทธิภาพในการทำงานลดลง เวลาที่ใช้ในการวัดประสิทธิภาพก็จะมากขึ้นตามไปด้วย

5.3.2 การเก็บรวบรวมไฟล์ในแต่ละหมวดหมู่ไฟล์พบว่า มีเนื้อหาหรือรายละเอียดของไฟล์ไม่ค่อยมีความแตกต่างกัน ส่งผลให้การวัดประสิทธิภาพการบีบอัดมีอัตราส่วนการบีบอัดที่ใกล้เคียงกัน

5.3.3 อัลกอริทึม Huffman Coding จะใช้เวลาในการประมวลผลการคลายข้อมูลนานมาก ถ้าหากไฟล์มีขนาดใหญ่ ตั้งแต่ 40 เมกะไบต์ขึ้นไป

5.4 ข้อเสนอแนะสำหรับการวิจัย

ผู้วิจัยมีข้อเสนอแนะบางประการเพื่อให้ผลการเปรียบเทียบการบีบอัดและโปรแกรมต้นแบบการบีบอัดข้อมูลมีประสิทธิภาพดียิ่งขึ้น ดังนี้

5.4.1 การคัดเลือกอัลกอริทึมที่จะนำมาใช้ในการวิจัย ต้องคำนึงถึงหมวดหมู่ของไฟล์ที่จะนำมาใช้ในทดสอบด้วย เนื่องจากอัลกอริทึมแต่ละตัว ถูกออกแบบและพัฒนาขีดความสามารถในการทำงานแตกต่างกัน มีความเหมาะสมกับข้อมูลไฟล์ที่แตกต่างกัน ทำให้มีผลกับการวัดประสิทธิภาพการทำงาน

5.4.2 ในการสร้างต้นแบบการบีบอัดข้อมูล ผู้วิจัยไม่มีการคัดเลือกหมวดหมู่ไฟล์จากไฟล์ที่นำเข้าสู่ระบบ หากมีการตรวจจับหมวดหมู่ไฟล์จากระบบอัตโนมัติ ก็จะทำให้การทำงานมีความสะดวกรวดเร็วมากยิ่งขึ้น

5.4.3 ในการวิจัยครั้งนี้ผู้วิจัยได้จำกัดประเภทไฟล์ข้อมูลเพื่อนำมาวัดผลการทดสอบและสร้างต้นแบบการบีบอัด หากผู้วิจัยต้องการนำต้นแบบไปใช้งานจริง จำเป็นต้องเพิ่มประเภทไฟล์อื่น ๆ เพื่อให้ครอบคลุมไฟล์ต่าง ๆ ที่มีอยู่ในปัจจุบันเข้าไปด้วย เพราะจะทำให้ระบบสามารถรองรับไฟล์ได้เพิ่มมากยิ่งขึ้น ก็จะทำให้ต้นแบบมีความสมบูรณ์แบบมากขึ้นตามไปด้วย

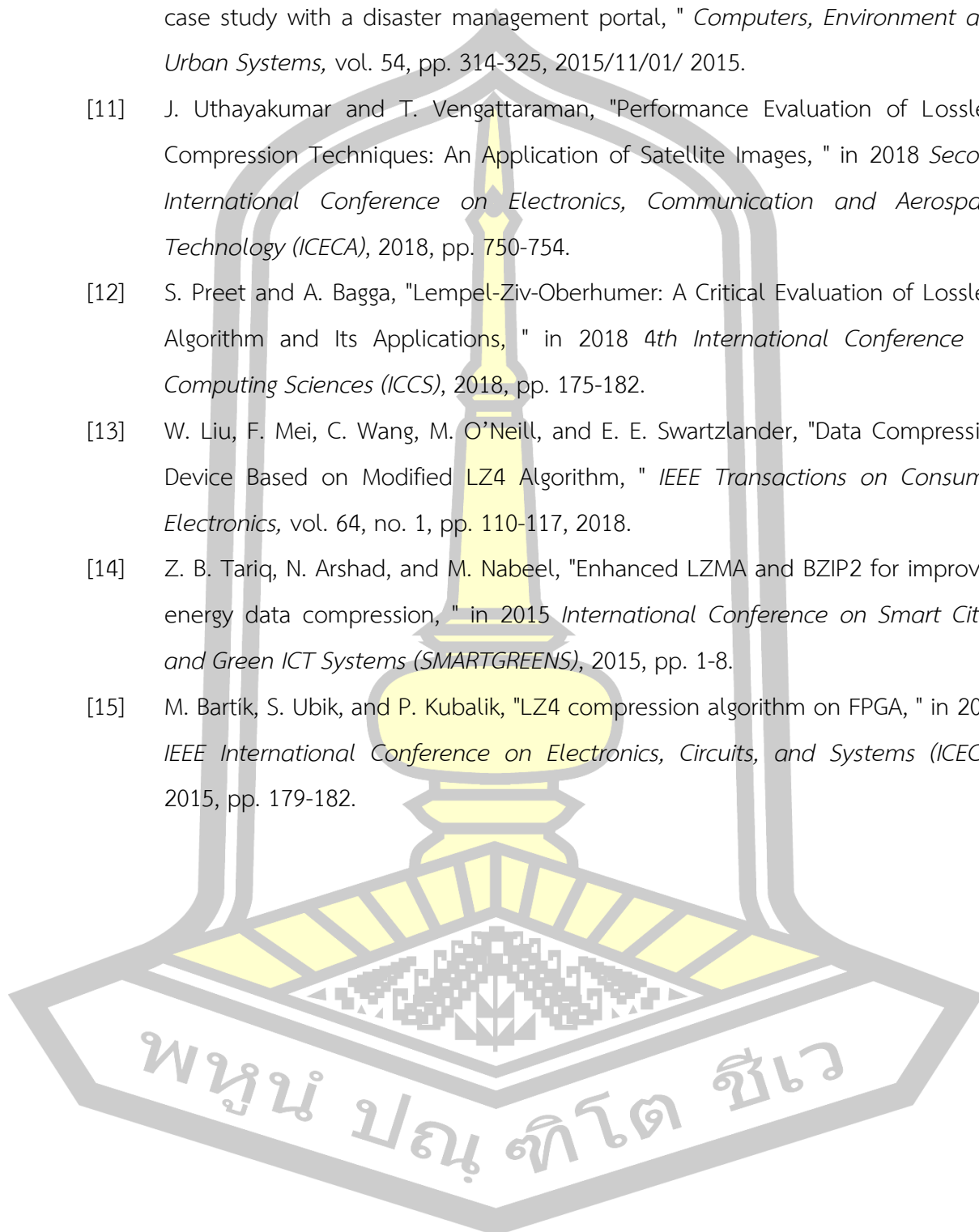
พหุ ประถมศึกษา ชีวะ

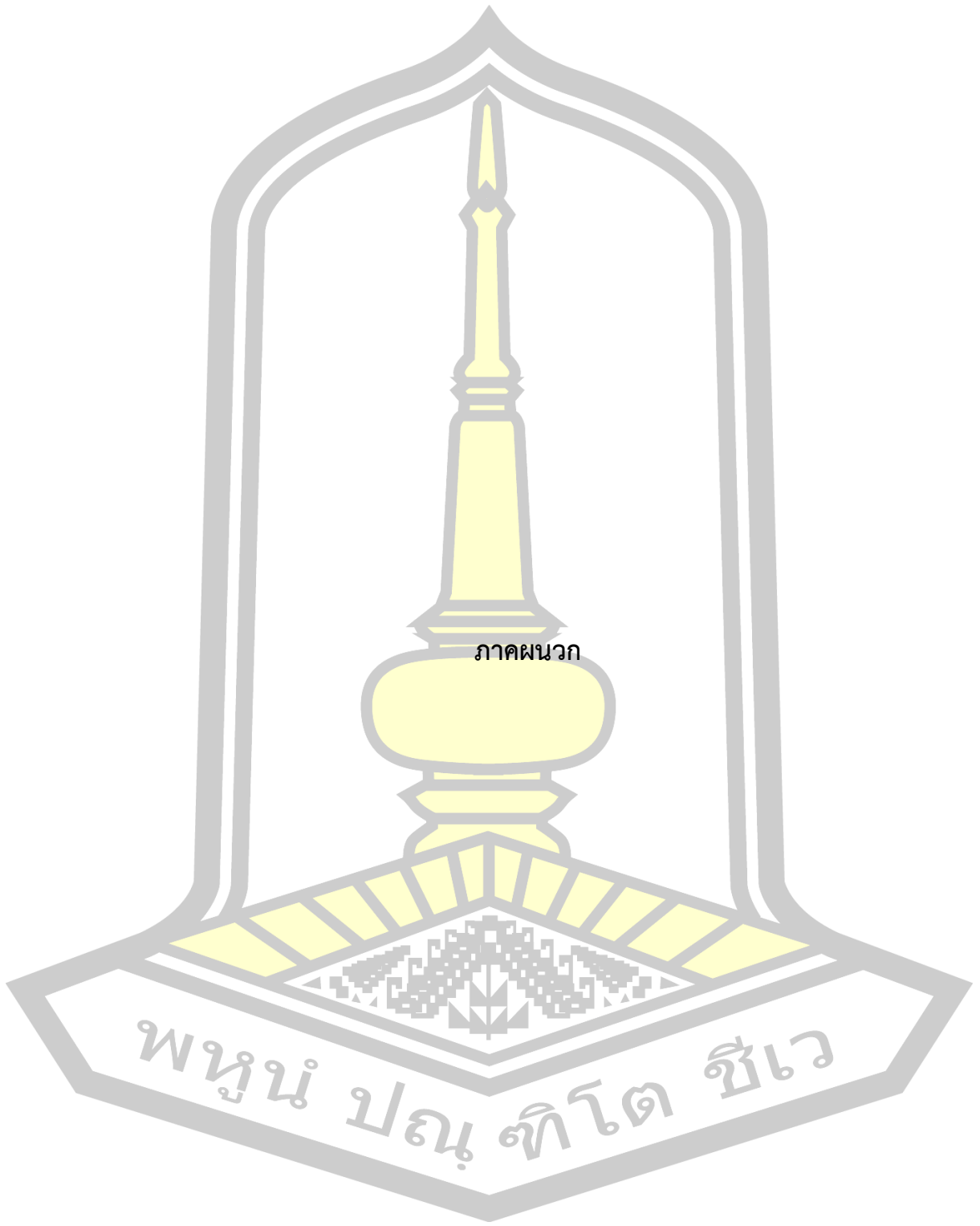
บรรณานุกรม

- [1] C. silawong and T. Anusamornkul, "A Comparative Study of Compression Algorithms for Each Data Type, " 2014.
- [2] ภควัฒน์ ติณสิริสุข, "การบีบอัดข้อมูลสำหรับเครือข่ายเซนเซอร์ไร้สาย, " 2559.
- [3] U. J, V. T, and D. P, "A new lossless neighborhood indexing sequence (NIS) algorithm for data compression in wireless sensor networks, " *Ad Hoc Networks*, vol. 83, pp. 149-157, 2019/02/01/ 2019.
- [4] X. Xudong and L. Yiran, "The application of LZMA algorithm in ISCS based on pretreatment, " in *2018 5th International Conference on Systems and Informatics (ICSAI)*, 2018, pp. 521-525.
- [5] J. Uthayakumar and T. Vengattaraman, "Compression Techniques are Lovable or Hateful: For Discrete Tone Images, " in *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 2018, pp. 744-749.
- [6] s. Pranveenit and K. Chanchio, "The Performance Analysis of Compression Techniques for Thread-Based Live Migration of Virtual Machine., " presented at the ICMSIT 2016 International Conference on Management Science, Innovation, and Technology 2016, 2016.
- [7] R. Jumar, H. Maaß, and V. Hagenmeyer, "Comparison of lossless compression schemes for high rate electrical grid time series for smart grid monitoring and analysis, " *Computers & Electrical Engineering*, vol. 71, pp. 465-476, 2018/10/01/ 2018.
- [8] M. B. Ibrahim and K. A. Gbolagade, "A Chinese Remainder Theorem Based Enhancements of Lempel-ziv-welch and Huffman Coding Image Compression, " *Asian Journal of Research in Computer Science*, pp. 1-9, 07/06 2019.
- [9] K. S. Kasmeeera, S. P. James, and K. Sreekumar, "Efficient Compression of Secured Images Using Subservient Data and Huffman Coding, " *Procedia Technology*, vol. 25, pp. 60-67, 2016/01/01/ 2016.
- [10] W. Li, M. Song, B. Zhou, K. Cao, and S. Gao, "Performance improvement

techniques for geospatial web services in a cyberinfrastructure environment – A case study with a disaster management portal, " *Computers, Environment and Urban Systems*, vol. 54, pp. 314-325, 2015/11/01/ 2015.

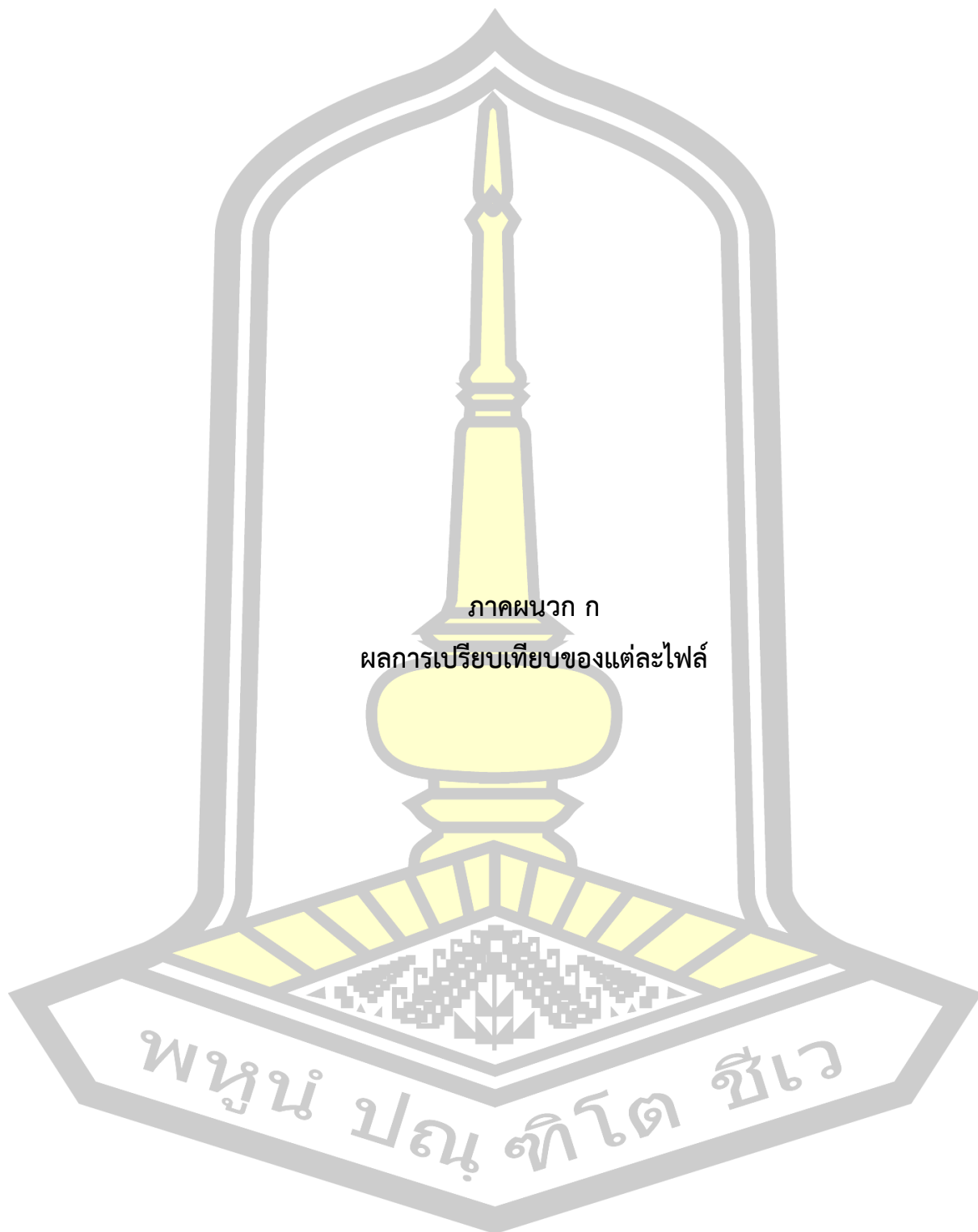
- [11] J. Uthayakumar and T. Vengattaraman, "Performance Evaluation of Lossless Compression Techniques: An Application of Satellite Images, " in *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 2018, pp. 750-754.
- [12] S. Preet and A. Bagga, "Lempel-Ziv-Oberhumer: A Critical Evaluation of Lossless Algorithm and Its Applications, " in *2018 4th International Conference on Computing Sciences (ICCS)*, 2018, pp. 175-182.
- [13] W. Liu, F. Mei, C. Wang, M. O'Neill, and E. E. Swartzlander, "Data Compression Device Based on Modified LZ4 Algorithm, " *IEEE Transactions on Consumer Electronics*, vol. 64, no. 1, pp. 110-117, 2018.
- [14] Z. B. Tariq, N. Arshad, and M. Nabeel, "Enhanced LZMA and BZIP2 for improved energy data compression, " in *2015 International Conference on Smart Cities and Green ICT Systems (SMARTGREENS)*, 2015, pp. 1-8.
- [15] M. Bartík, S. Ubik, and P. Kubalik, "LZ4 compression algorithm on FPGA, " in *2015 IEEE International Conference on Electronics, Circuits, and Systems (ICECS)*, 2015, pp. 179-182.





ภาคผนวก

พหุมนุ ปณู ทิโต ชีเว



ภาคผนวก ก
ผลการเปรียบเทียบของแต่ละไฟล์

พหุบัน ปณฺ ทั โตะ ชีเว

1. ผลการเปรียบเทียบอัตราความเร็วการบีบอัด (Compression speed) แยกเป็นไฟล์

ชื่อไฟล์	อัลกอริทึม (อัตราส่วน : 1 ล้านไบต์ต่อวินาที)				
	BZip2	DEFLATE	Huffman Coding	LZ4	LZMA
หมวด : Group_Data					
000004.log	1.75	72.68	3.81	64.58	1.67
101_Full.pdf	0.87	23.10	2.35	78.09	0.74
120Page.pdf	0.91	20.46	2.42	47.20	0.82
132079212448625110.pdf	0.89	24.34	2.48	180.92	0.80
2.pdf	1.02	23.26	2.48	38.64	0.88
3.pdf	0.89	21.07	2.71	76.76	0.83
5000568.pdf	0.84	21.86	2.45	172.80	0.74
5000664.pdf	0.89	22.38	2.46	186.53	0.74
5001157.pdf	0.89	23.38	2.48	174.08	0.84
5001797.pdf	0.93	23.36	2.57	189.51	0.85
5100713.pdf	0.81	21.24	2.55	179.70	0.74
5100854.pdf	0.96	22.36	2.48	145.46	0.86
5707319.pdf	0.95	23.74	2.50	132.24	0.86
5707358.pdf	0.93	23.46	2.46	154.10	0.80
5707901.pdf	0.95	22.45	2.36	82.31	0.82
570790189.pdf	1.05	23.98	2.26	70.45	0.74
61080810.pdf	1.09	24.64	2.34	44.55	0.86
Append.pdf	1.02	23.69	2.51	48.39	1.56
ASPxPivotGrid.pdf	0.98	25.50	2.48	61.72	0.83
bch01.log	0.42	1.29	0.73	1.22	0.31
Book1.xlsx	1.17	22.15	2.39	25.21	0.65
Book1d.xlsx	1.10	23.25	2.36	22.46	0.63
bootstrap_log.txt	0.97	3.96	0.78	3.86	0.34
CRM.xlsx	1.14	23.02	2.28	27.25	0.62
DataConnections.xlsx	1.12	22.28	2.34	24.09	0.63
DB_EDocuments_backup_2018_03_28_070000_9724092.bak	4.77	63.72	3.79	55.37	0.94
DB_EDocuments_Full_TranserEDoc_Completed.fbak	2.92	49.40	2.97	43.02	0.66
DPINST.LOG	0.33	1.76	0.40	1.47	0.17
h.pdf	1.05	23.37	2.43	32.39	0.80
large_empty.scss.txt	28.30	553.32	4.96	406.80	2.90
ngen.log	2.86	74.40	3.02	69.86	1.54

1. ผลการเปรียบเทียบอัตราความเร็วการบีบอัด (Compression speed) แยกเป็นไฟล์

ชื่อไฟล์	อัลกอริทึม (อัตราส่วน : 1 ล้านไบต์ต่อวินาที)				
	BZip2	DEFLATE	Huffman Coding	LZ4	LZMA
OKTest.pdf	1.16	22.27	2.53	83.53	0.90
PFR0.log	0.78	9.99	2.13	15.20	0.86
project-33.doc	1.22	26.52	2.51	58.09	0.90
Propersal.doc	1.76	16.42	2.56	15.51	0.79
Propersal_ส่งหัวเรื่องแล้ว.docx	0.82	8.50	1.82	9.34	0.65
setupapi.offline.20190318_215229.log	3.52	106.79	3.06	102.40	1.78
setupapi.offline.log	1.05	23.22	2.39	45.17	1.12
setupapi.upgrade.log	1.06	23.94	2.45	46.07	0.98
Spect TOR CallCenter.xlsx	0.58	2.00	0.89	1.95	0.33
svnserve.bak	0.56	2.00	0.48	1.88	0.18
WindowsUpdate.log	0.07	0.12	0.03	0.12	0.01
การบีบอัดข้อมูลในเครือข่าย(1).pdf	1.08	25.14	2.44	62.64	0.94
การบีบอัดเอกสารเอชทีเอ็มแอลบนฝั่งเซิร์ฟเวอร์ด้วยการเข้ารหัสแบบฮัฟแมน(1).pdf	1.06	20.15	2.40	35.18	1.00
การพัฒนาอัลกอริทึมการบีบอัดแบบไม่สูญเสียข้อมูลบนเว็บแอปพลิเคชัน_New.docx	1.07	22.66	2.36	37.65	1.01
การศึกษาเปรียบเทียบวิธีบีบอัดข้อมูลที่เหมาะสม - 1.pdf	1.19	20.08	2.39	26.28	1.04
คลังร้อยเอ็ด.txt	0.33	0.97	0.17	0.85	0.11
คำอธิบาย LZMA.pdf	0.94	20.96	2.37	30.46	0.92
คู่มือการใช้งาน SSIS.pdf	0.93	21.90	2.32	43.54	0.84
ตัวอย่างระบบ CallIncome.doc	1.02	21.60	2.51	28.48	0.90
เทคนิคการสืบค้นสารสนเทศและทำเหมืองข้อความ.pdf	0.92	23.63	2.49	37.42	0.80
ไบเสนราคา.doc	1.58	6.12	1.83	6.22	0.83
สูตรการตั้งสำรอง.xlsx	0.98	19.57	2.42	62.25	0.99
หมวด : Group_Images					
001_55.tif	1.66	3.96	1.52	3.45	0.73
001_60.tif	1.80	8.38	1.39	3.46	0.65
58965.jpg	1.13	25.29	2.39	108.94	0.97
9856.tif	1.73	24.34	2.47	24.02	0.55
add-01.psd	1.84	24.79	1.77	16.03	0.97
BAY090-0001.tif	1.73	22.64	2.75	20.00	0.57
BAY090-0001-0003.tif	1.81	23.55	2.62	19.87	0.54

1. ผลการเปรียบเทียบอัตราความเร็วการบีบอัด (Compression speed) แยกเป็นไฟล์

ชื่อไฟล์	อัลกอริทึม (อัตราส่วน : 1 ล้านไบต์ต่อวินาที)				
	BZip2	DEFLATE	Huffman Coding	LZ4	LZMA
BAY127.jpg	1.16	23.24	1.95	166.13	0.89
BAY133.png	1.02	24.66	2.60	171.36	0.93
BAY133.psd	1.22	23.24	2.09	72.07	0.85
BAY141.gif	0.99	25.56	2.38	161.19	0.90
BAY294.JPG	1.06	22.47	2.43	92.30	1.03
binoculars-01.psd	1.45	27.33	2.66	17.59	0.95
Cancel.gif	0.20	0.43	0.10	0.61	0.07
CD.gif	0.16	0.19	0.08	0.19	0.07
delicious.psd	2.11	31.02	2.82	25.03	0.87
DesktopKeepOnToastimg.gif	1.10	15.44	2.19	20.26	0.91
KTB586.JPG	1.09	21.76	2.38	73.35	1.04
KTB638.JPG	1.10	22.30	2.41	92.86	1.03
KTB656.JPG	1.08	22.07	2.41	98.56	1.03
KTB793.JPG	1.10	21.41	2.33	64.17	1.05
logo.png	0.51	1.58	0.74	1.58	0.38
mail.psd	2.16	33.18	2.80	27.61	0.86
mAlNn.psd	1.70	23.49	2.47	34.06	0.63
Musics.png	1.01	20.21	2.52	60.27	1.02
noimage.png	0.35	0.96	0.39	2.82	0.24
Office Publisher.png	0.96	14.61	2.17	25.61	0.98
ok.psd	1.73	24.22	2.54	22.81	0.54
ok.tif	2.36	33.74	2.78	31.86	0.70
OnInstallSnapshot.png	1.08	20.07	2.33	77.66	1.00
oobe-bookend-cortanain.gif	0.98	10.30	1.81	12.90	0.93
Page1_WebChartControl1_ChartsImage.bmp	5.74	86.50	3.54	108.62	2.73
PageNotFound.png	0.80	4.94	1.45	5.22	0.73
Recycle-Bin-Basic-Empty.png	1.00	15.87	2.34	41.90	1.01
reminder.gif	1.23	13.67	2.05	15.31	1.09
s.png	1.01	21.14	2.53	60.81	1.00
sprite.gif	0.46	0.88	0.46	2.00	0.22
TextBox สำหรับ Username.jpg	0.21	0.81	0.14	0.83	0.08
WelcomeFax.tif	1.33	10.33	1.97	10.45	1.04

1. ผลการเปรียบเทียบอัตราความเร็วการบีบอัด (Compression speed) แยกเป็นไฟล์

ชื่อไฟล์	อัลกอริทึม (อัตราส่วน : 1 ล้านไบต์ต่อวินาที)				
	BZip2	DEFLATE	Huffman Coding	LZ4	LZMA
win7-desktop-green-bg.jpg	1.10	20.97	2.45	75.94	1.02
YaaHoo.psd	1.81	22.42	2.53	14.72	0.00
ZPLDZ.psd	1.85	25.01	2.48	16.98	0.50
หมวด : Group_Music					
ก็มาติคะ - ยุ่งยุ่ง กนกนันท์ feat.Night tingle (OFFICIAL MV)(0).mp4	0.96	24.38	2.27	169.55	0.87
ขอบใจเด้อ - ศาล สานศิลป์ เจิ้ง Music.mp3	0.94	23.12	2.21	127.82	0.96
ขอโอกาสแทนเด้อ - บอย พนมไพร.mp3	1.06	23.76	2.46	122.82	0.98
คนละฟ้า - แจ็ค ลูกอีสาน (Official Music Video).mp3	0.95	20.77	2.36	99.88	0.94
คืนมาสาหล่า - แจ็คลูกอีสาน (Official MV).mp3	1.05	22.72	2.52	115.68	0.94
คืนมาสาหล่า - แจ็คลูกอีสาน (Official MV).wav	1.09	20.42	2.34	131.93	0.79
แจ๊คลูกอีสาน คำแม่แจ๊ค ลูกอีสาน.mp3	1.06	23.72	2.25	86.89	0.93
เด้อเที่ยงต่าง ไ ล้ำโย ไททองคำ.wmv	1.08	21.30	2.45	37.29	0.86
เด้อเที่ยงต่าง.wmv	1.18	24.71	2.25	47.61	0.88
นอนสาหล่า - สมาร์ท วาทยโยธา.mp4	0.97	24.19	2.26	155.41	0.95
นอนสาหล่า.avi	0.97	23.63	2.43	164.80	0.83
นอนสาหล่า.mp4	0.95	24.32	2.52	184.97	0.89
บั้งจักเม็ด.wmv	0.96	21.18	2.39	46.76	0.75
บัดนี้ - แจ๊ค ลูกอีสาน (Official MV).wav	1.01	22.31	2.38	188.66	0.82
ปี่(จนป่น - เอ มหาหงส์ feat.บัว กมลทิพย์ Cover MV น่องน้ำฝน พืด).mp4	0.92	23.76	2.43	175.74	0.89
เป้ ภาณุชัย พรสวรรค์ ท็อปไลน์EP.1.mp3	0.98	22.95	2.37	141.33	0.82
ผู้สาวมักอ้ายบ่ - น่องเอเซีย [Cover MV] น่องอินดี้ น่องนุ่น บะเคซิติ้.mp4	0.94	24.60	2.46	200.72	0.87
ฝนเพลงมา - กาเน็ต สะเลอปี [Cover MV] น่องอินดี้ ทีมเด่น บะเคซิติ้.avi	0.98	24.18	2.47	87.62	0.85
ฝนเพลงมา - กาเน็ต สะเลอปี [Cover MV] น่องอินดี้ ทีมเด่น บะเคซิติ้.mp4	0.89	21.73	2.43	185.73	0.76
ฝนเพลงมา - กาเน็ต สะเลอปี [Cover MV] น่องอินดี้ ทีมเด่น บะเคซิติ้.wmv	1.07	23.63	2.47	38.28	0.86
ฝนเพลงมา - กาเน็ต สะเลอปี น่องอินดี้ ทีมเด่น บะเคซิติ้.avi	1.03	23.73	2.24	78.30	0.83
พ้อค่าน้อย.mp4	0.94	24.74	2.43	201.96	0.87
พ้อค่าน้อย.wmv	0.99	22.97	2.40	56.00	0.85
ฟังเพลิน3ชม ขับรถ, ทำงาน, บุญบ้าน, เอ อรุษา กุ้ง สุภาพร.mp3	0.90	21.86	2.38	141.06	0.73
ฟังเพลิน3ชม ขับรถ, ทำงาน, บุญบ้าน, เอ อรุษา กุ้ง สุภาพร.wav	1.09	21.00	2.53	189.28	0.71
รถแห่รถยู๊ (ปี๋โล้งโป้งฉิ่ง) - น่อง ทิวθεν Cover MV โดย น่องโปรแกรม(0).mp4	0.92	22.31	2.45	176.84	0.83
รถแห่รถยู๊ (ปี๋โล้งโป้งฉิ่ง) - น่อง ทิวθεν Cover MV โดย น่องโปรแกรม.mp4	0.95	24.42	2.36	189.25	0.88
รถแห่รถยู๊ (ปี๋โล้งโป้งฉิ่ง) - น่อง ทิวθεν Cover MV โดย น่องโปรแกรม.wmv	1.02	23.14	2.21	54.07	0.84

1. ผลการเปรียบเทียบอัตราความเร็วการบีบอัด (Compression speed) แยกเป็นไฟล์

ชื่อไฟล์	อัลกอริทึม (อัตราส่วน : 1 ล้านไบต์ต่อวินาที)				
	BZip2	DEFLATE	Huffman Coding	LZ4	LZMA
รวมเพลง มนต์แคน แก่นคูน 15 เพลงฮิตจากบ่าวเสียงสุดสะแนน.mp3	0.97	23.11	2.12	165.31	0.84
รวมเพลง มนต์แคน แก่นคูน 15 เพลงฮิตจากบ่าวเสียงสุดสะแนน.wav	1.07	23.45	2.38	201.07	0.86
รวมเพลงลูกทุ่งเจ็บ ๆ โดน ๆ ซึ้ง ๆ ฟังแล้วน้ำตาจไหล ยอดฮิตใหม่ล่าสุด2019.mp3	0.98	23.29	2.37	124.56	0.85
รวมเพลงฮิต ผู้บ่าวไทบ้าน ชูปดาร์อีสาน l หนุ่ม น.ป.ช. , อส. รอรัก , ดาวมีไว้เบิ่ง.mp3	0.97	22.88	2.14	154.24	0.84
รักติดไซเรน (My Ambulance) - ไอซ์ พาริส, แพรวา ณิชากัทร [Official MV] Nadao Music.avi	1.02	23.58	2.36	114.80	0.84
หมอลำล่งทุ่ง ลงนา.mp3	1.04	20.58	2.49	13.92	0.68
ลูกก้า.avi	1.00	23.80	2.21	165.40	0.84
เอ อนุชา พาย้อนยุค 10 ปีคืนหลัง ฟังลำซิ่ง1990.mp3	0.99	23.16	2.40	24.57	0.79
เอ อนุชา พาย้อนยุค 10 ปีคืนหลัง ฟังลำซิ่ง1990.wav	1.42	23.21	2.39	150.55	0.62
เอาอย่างจ้.mp4	0.88	23.42	2.31	202.05	0.80
Total	1.38	25.98	2.26	77.26	0.82

2. ผลการเปรียบเทียบอัตราความเร็วการคลายข้อมูล (Decompression speed) แยกเป็นไฟล์

ชื่อไฟล์	อัลกอริทึม (อัตราส่วน : 1 ล้านไบต์ต่อวินาที)				
	BZip2	DEFLATE	Huffman Coding	LZ4	LZMA
หมวด : Group_Data					
000004.log	0.56	16.78	0.08	27.86	0.75
101_Full.pdf	3.01	48.88	0.08	676.12	2.80
120Page.pdf	3.43	41.50	0.09	504.74	2.99
132079212448625110.pdf	3.38	422.82	0.07	853.47	2.89
2.pdf	3.14	59.08	0.08	224.59	2.85
3.pdf	3.47	71.17	0.08	230.66	2.93
5000568.pdf	3.59	219.73	0.08	758.36	2.79
5000664.pdf	3.68	450.93	0.08	819.22	2.96
5001157.pdf	3.58	420.79	0.08	920.95	3.10
5001797.pdf	3.59	182.04	0.09	1, 044.96	3.04
5100713.pdf	3.30	271.61	0.08	875.29	3.11
5100854.pdf	3.61	116.85	0.09	608.66	3.00
5707319.pdf	3.57	239.19	0.09	572.91	3.01

2. ผลการเปรียบเทียบอัตราความเร็วการคลายข้อมูล (Decompression speed) แยกเป็นไฟล์

ชื่อไฟล์	อัลกอริทึม (อัตราส่วน : 1 ล้านไบต์ต่อวินาที)				
	BZip2	DEFLATE	Huffman Coding	LZ4	LZMA
5707358.pdf	3.63	355.08	0.09	893.51	3.00
5707901.pdf	3.19	154.92	0.08	462.45	2.94
570790189.pdf	3.49	140.99	0.08	378.65	2.89
61080810.pdf	3.49	42.25	0.09	406.78	3.03
Append.pdf	3.57	63.47	0.09	198.13	2.37
ASPxPivotGrid.pdf	3.16	92.56	0.09	200.63	2.82
bch01.log	0.97	3.04	0.11	5.57	1.09
Book1.xlsx	2.77	46.88	0.08	482.12	3.13
Book1d.xlsx	2.67	58.29	0.09	372.19	3.08
bootstrap_log.txt	0.36	0.78	0.11	1.40	0.21
CRM.xlsx	2.78	37.03	0.09	599.66	3.15
DataConnections.xlsx	2.63	45.51	0.08	483.66	2.85
DB_EDocuments_backup_2018_03_28_070000_9724092.bak	1.29	18.57	0.08	33.05	1.78
DB_EDocuments_Full_TranserEDoc_Completed.fbak	0.97	27.09	0.08	65.09	1.75
DPINST.LOG	0.38	0.58	0.13	0.87	0.46
h.pdf	2.97	66.60	0.09	260.39	2.92
large_empty.scss.txt	0.00	2.73	0.01	2.63	0.01
ngen.log	0.34	7.77	0.09	9.28	0.86
OKTest.pdf	2.92	53.26	0.08	229.13	3.06
PFRO.log	3.22	33.63	0.09	49.81	2.46
project-33.doc	2.79	66.09	0.08	242.67	2.72
Propersal.doc	2.05	14.59	0.09	20.63	2.04
Propersal_ส่งหัวเรื่องแล้ว.docx	2.68	25.22	0.09	20.76	2.26
setupapi.offline.20190318_215229.log	0.49	12.98	0.08	26.69	1.54
setupapi.offline.log	3.64	53.80	0.08	127.96	2.87
setupapi.upgrade.log	3.48	53.58	0.08	236.02	2.89
Spect TOR CallCenter.xlsx	1.10	3.44	0.10	4.77	1.08
svnservice.bak	0.50	0.81	0.15	1.29	0.56
WindowsUpdate.log	0.14	0.11	0.28	0.18	0.09
การบีบอัดข้อมูลในเครือข่าย(1).pdf	3.29	74.29	0.09	223.21	2.91
การบีบอัดเอกสารเอชทีเอ็มแอลบนฝั่งเซิร์ฟเวอร์ ด้วยการเข้ารหัสแบบฮิปแมน(1).pdf	3.46	54.96	0.09	107.55	2.84
การพัฒนาอัลกอริทึมการบีบอัดแบบไม่สูญเสีย	3.21	53.88	0.07	149.60	2.88

2. ผลการเปรียบเทียบอัตราความเร็วการคลายข้อมูล (Decompression speed) แยกเป็นไฟล์

ชื่อไฟล์	อัลกอริทึม (อัตราส่วน : 1 ล้านไบต์ต่อวินาที)				
	BZip2	DEFLATE	Huffman Coding	LZ4	LZMA
ข้อมูลบนเว็บแอปพลิเคชัน_New.docx					
การศึกษาเปรียบเทียบวิธีบีบอัดข้อมูลที่เหมาะสม - 1.pdf	3.22	46.60	0.09	74.38	2.70
คลังร้อยเอ็ด.txt	0.17	0.15	0.21	0.14	0.11
คำอธิบาย LZMA.pdf	3.42	51.55	0.09	102.79	2.83
คู่มือการใช้งาน SSIS.pdf	3.37	59.39	0.09	202.43	2.91
ตัวอย่างระบบ CallIncome.doc	3.05	65.11	0.08	140.47	2.90
เทคนิคการสืบค้นสารสนเทศและทำเหมืองข้อความ.pdf	3.41	63.00	0.09	273.06	2.95
ใบเสนอราคา.doc	0.88	3.38	0.10	3.53	0.83
สูตรการตั้งสำรอง.xlsx	3.28	54.30	0.08	150.68	3.02
หมวด : Group_Images					
001_55.tif	0.84	2.38	0.08	2.81	0.88
001_60.tif	0.84	2.35	0.08	2.97	0.59
58965.jpg	1.63	65.04	0.08	230.55	2.04
9856.tif	1.22	26.55	0.08	84.57	1.71
add-01.psd	1.05	31.25	0.07	102.13	1.59
BAY090-0001.tif	1.20	40.91	0.08	149.12	1.49
BAY090-0001-0003.tif	1.19	39.80	0.08	144.75	1.51
BAY127.jpg	1.63	63.35	0.07	296.22	1.96
BAY133.png	1.70	249.71	0.07	273.33	2.04
BAY133.psd	1.64	101.17	0.07	252.49	1.85
BAY141.gif	1.67	246.95	0.08	299.67	2.02
BAY294.JPG	1.71	174.05	0.07	156.89	2.06
binoculars-01.psd	0.98	36.91	0.07	116.26	1.48
Cancel.gif	0.59	0.42	0.17	0.40	0.31
CD.gif	0.42	0.30	0.24	0.24	0.22
delicious.psd	0.91	31.06	0.07	58.27	1.45
DesktopKeepOnToastImg.gif	1.63	41.48	0.08	76.26	1.99
KTB586.JPG	1.73	121.67	0.07	200.05	2.01
KTB638.JPG	1.75	147.78	0.07	212.38	2.00
KTB656.JPG	1.77	97.68	0.08	235.27	2.00
KTB793.JPG	1.76	121.57	0.07	133.46	2.04
logo.png	1.32	3.69	0.08	3.90	1.25
mail.psd	0.88	31.71	0.08	55.09	1.45

2. ผลการเปรียบเทียบอัตราความเร็วการคลายข้อมูล (Decompression speed) แยกเป็นไฟล์

ชื่อไฟล์	อัลกอริทึม (อัตราส่วน : 1 ล้านไบต์ต่อวินาที)				
	BZip2	DEFLATE	Huffman Coding	LZ4	LZMA
mAINn.psd	1.38	43.43	0.08	131.75	1.77
Musics.png	1.56	75.29	0.08	133.31	1.96
noimage.png	0.92	2.00	0.08	2.02	0.96
Office Publisher.png	1.74	38.66	0.08	52.35	2.04
ok.psd	1.17	20.15	0.08	109.14	1.84
ok.tif	1.15	30.52	0.08	85.44	1.66
OnInstallSnapshot.png	1.65	52.08	0.07	169.35	1.96
oobe-bookend-cortanain.gif	1.88	25.31	0.07	30.47	1.95
Page1_WebChartControl1_ChartsImage.bmp	0.06	4.71	0.08	4.68	0.29
PageNotFound.png	1.52	12.22	0.08	14.40	1.67
Recycle-Bin-Basic-Empty.png	1.84	76.65	0.08	78.20	2.04
reminder.gif	1.47	22.17	0.07	33.79	1.81
s.png	1.70	80.55	0.07	174.57	2.08
sprite.gif	0.97	2.34	0.10	1.95	0.92
TextBox สำหรับ Username.jpg	0.70	0.62	0.17	0.56	0.40
WelcomeFax.tif	1.46	17.77	0.07	18.98	1.76
win7-desktop-green-bg.jpg	1.76	61.21	0.08	178.07	2.04
YaaHoo.psd	1.02	8.87	0.08	9.01	0.00
ZPLDZ.psd	1.05	41.01	0.07	114.62	1.87
หมวด : Group_Music					
ก๊ิมาดิคะ - ยุงยั้ง กนกนันทน์ feat.Night tingle (OFFICIAL MV)(0).mp4	3.85	183.68	0.07	875.61	3.06
ขอบใจเต๋อ - ศาล สานศิลป์ เจริญ Music.mp3	3.74	46.89	0.08	426.91	3.08
ขอโอกาสแทนเต๋อ - บอย พนมไพร.mp3	3.18	56.75	0.08	209.50	2.76
คนละฟ้า - แจ็ค ลูกอีสาน (Official Music Video).mp3	3.62	54.53	0.08	370.38	2.81
คืนมาสาหล่า - แจ็คลูกอีสาน (Official MV).mp3	3.50	61.40	0.08	439.51	3.02
คืนมาสาหล่า - แจ็คลูกอีสาน (Official MV).wav	3.42	38.81	0.07	771.89	2.93
แจ๊คลูกอีสาน คำแม่แจ๊ค ลูกอีสาน.mp3	3.74	49.43	0.08	446.89	3.11
เต๋อเต๋อต่าง เ ล้าโย ไททองคำ.wmv	3.57	45.61	0.08	325.70	2.71
เต๋อเต๋อต่าง.wmv	3.37	53.74	0.08	357.33	3.07
นอนสาหล่า - สมาร์ท วาทยธนา.mp4	3.84	104.01	0.08	383.19	3.04
นอนสาหล่า.avi	3.84	98.80	0.08	933.10	2.97
นอนสาหล่า.mp4	3.82	416.60	0.08	767.55	3.00
บั้งจักเม็ด.wmv	3.48	39.96	0.08	626.99	3.02

2. ผลการเปรียบเทียบอัตราความเร็วการคลายข้อมูล (Decompression speed) แยกเป็นไฟล์

ชื่อไฟล์	อัลกอริทึม (อัตราส่วน : 1 ล้านไบต์ต่อวินาที)				
	BZip2	DEFLATE	Huffman Coding	LZ4	LZMA
บัดนี่ - แจ็ค ลูกอีสาน (Official MV).wav	3.64	44.42	0.08	925.31	2.99
ปี่(จน)ป่น - เอ มหาหงส์ feat.บัว กมลทิพย์ Cover MV น่องน้ำฝน พี้ตี.mp4	3.59	423.44	0.08	883.17	3.08
เป้ ภาณุชัย พรสวรรค์ ท็อปไลน์EP.1.mp3	3.53	62.22	0.07	1, 023.59	3.00
ผู้สาวมักอ้ายบ่ - น่องเอเชีย [Cover MV] น่องอินดี้ น่องนุ่น บะเคซิติ้.mp4	3.68	710.24	0.90	956.64	3.03
ฝนเทลงมา - กาเน็ต สะเลอปปี้ [Cover MV] น่องอินดี้ ทิมเต้น บะเคซิติ้.avi	3.66	79.06	0.03	643.90	3.12
ฝนเทลงมา - กาเน็ต สะเลอปปี้ [Cover MV] น่องอินดี้ ทิมเต้น บะเคซิติ้.mp4	3.72	741.86	2.24	1, 009.40	3.04
ฝนเทลงมา - กาเน็ต สะเลอปปี้ [Cover MV] น่องอินดี้ ทิมเต้น บะเคซิติ้.wmv	3.67	53.55	0.02	426.63	3.09
ฝนเทลงมา - กาเน็ต สะเลอปปี้ น่องอินดี้ ทิมเต้น บะเคซิติ้.avi	3.47	52.45	0.07	823.27	3.01
พ้อค่าน้อย.mp4	3.72	835.42	0.07	964.74	2.96
พ้อค่าน้อย.wmv	3.70	50.15	0.08	726.18	2.97
ฟังเพลิน3ชม ขับรถ, ทำงาน , บุญบ้าน, เอ อนุชา กุ้ง สุภาพร.mp3	3.71	42.06	0.08	968.35	3.06
ฟังเพลิน3ชม ขับรถ, ทำงาน , บุญบ้าน, เอ อนุชา กุ้ง สุภาพร.wav	3.17	37.12	0.08	1, 033.69	2.86
รถแห่รถยู๊ (ปีะโล้งโป้งฉิ่ง) - น่อง ทิวเทน Cover MV โดย น่องโปรแกรม(0).mp4	3.51	292.11	0.08	934.07	3.00
รถแห่รถยู๊ (ปีะโล้งโป้งฉิ่ง) - น่อง ทิวเทน Cover MV โดย น่องโปรแกรม.mp4	3.74	365.23	0.07	946.27	3.04
รถแห่รถยู๊ (ปีะโล้งโป้งฉิ่ง) - น่อง ทิวเทน Cover MV โดย น่องโปรแกรม.wmv	3.74	42.98	0.07	709.02	2.94
รวมเพลง มนต์แคน แก่นคูน 15 เพลงฮิตจากบ่าวเสียงสุดสะแนน.mp3	3.75	61.30	0.07	979.90	3.01
รวมเพลง มนต์แคน แก่นคูน 15 เพลงฮิตจากบ่าวเสียงสุดสะแนน.wav	3.54	42.84	0.07	950.22	2.98
รวมเพลงลูกทุ่งเจ็บ ๆ โดน ๆ ซึ้ง ๆ ฟังแล้วน้ำตาจิไหล ยอดฮิตใหม่ล่าสุด2019.mp3	3.61	43.83	0.07	941.72	3.07
รวมเพลงฮิต ผู้บ่าวไทบ้าน ชูปดาร์อีสาน l หนุ่ม น.ป.ช. , อส. รอรัก , ดาวมีไว้เบิ่ง.mp3	3.66	44.05	0.07	913.76	3.05
รักติดไซเรน (My Ambulance) - ไอซ์ พาริส, แพรวา นิชาภัทร [Official MV] Nadao Music.avi	3.65	58.90	0.75	906.01	3.06
หมอลำล่งทุ่ง ลงนา.mp3	3.33	64.53	0.08	427.08	3.06
อุกก้า.avi	3.79	84.34	0.07	970.95	2.98
เอ อนุชา พาย้อนยุค 10 ปีคืนหลัง ฟังลำซิ่ง1990.mp3	3.76	57.08	0.08	542.54	3.05
เอ อนุชา พาย้อนยุค 10 ปีคืนหลัง ฟังลำซิ่ง1990.wav	2.50	32.59	0.08	947.43	3.30
เอาอย่างจ้.mp4	3.53	715.19	0.08	1, 009.87	3.02
Total	2.43	97.02	0.11	366.46	2.32

3. ผลการเปรียบเทียบอัตราส่วนการบีบอัดข้อมูล (Compression Ratio) แยกเป็นไฟล์

ชื่อไฟล์	อัลกอริทึม (อัตราส่วน : ร้อยละขนาดไฟล์)				
	BZip2	DEFLATE	Huffman Coding	LZ4	LZMA
หมวด : Group_Data					
000004.log	4.64	8.53	42.97	13.10	1.37
101_Full.pdf	94.77	95.19	99.87	96.25	94.01
120Page.pdf	89.81	90.59	98.96	91.79	89.40
132079212448625110.pdf	100.24	99.87	100.02	99.89	96.04
2.pdf	90.04	90.59	99.09	92.22	89.69
3.pdf	97.51	97.22	100.08	97.96	97.54
5000568.pdf	100.25	99.85	100.02	99.93	100.49
5000664.pdf	100.38	99.97	100.02	99.98	100.77
5001157.pdf	100.40	100.01	100.02	100.00	100.82
5001797.pdf	100.34	99.91	100.02	99.99	100.64
5100713.pdf	100.31	99.90	100.02	99.97	100.57
5100854.pdf	99.89	99.38	100.03	99.71	99.68
5707319.pdf	99.65	99.32	100.03	99.74	99.99
5707358.pdf	99.76	99.47	100.02	99.81	100.29
5707901.pdf	96.66	97.07	100.02	98.74	97.07
570790189.pdf	94.98	95.83	99.98	98.16	95.43
61080810.pdf	87.62	88.65	98.44	89.91	87.71
Append.pdf	88.47	88.38	98.46	89.12	22.14
ASPxPivotGrid.pdf	79.79	79.61	97.83	82.10	78.71
bch01.log	101.06	94.52	127.11	95.68	95.11
Book1.xlsx	84.62	90.19	99.77	97.79	80.01
Book1d.xlsx	84.40	89.30	99.67	97.34	80.30
bootstrap_log.txt	20.75	17.04	86.82	24.89	16.54
CRM.xlsx	84.09	90.71	99.92	98.14	78.91
DataConnections.xlsx	85.18	89.84	99.84	97.56	81.04
DB_EDocuments_backup_2018_03_28_070000_9724092.bak	10.12	12.00	41.20	17.73	7.35
DB_EDocuments_Full_TranserEDoc_Completed.fbak	10.88	15.68	67.02	23.44	8.32
DPINST.LOG	30.14	32.66	110.71	54.46	29.51
h.pdf	86.09	87.74	99.61	90.05	84.44
large_empty.scss.txt	0.00	0.43	12.53	0.41	0.01
ngen.log	3.14	4.70	66.91	7.19	3.92
OKTest.pdf	90.97	95.14	99.01	98.45	92.50

3. ผลการเปรียบเทียบอัตราส่วนการบีบอัดข้อมูล (Compression Ratio) แยกเป็นไฟล์

ชื่อไฟล์	อัลกอริทึม (อัตราส่วน : ร้อยละขนาดไฟล์)				
	BZip2	DEFLATE	Huffman Coding	LZ4	LZMA
PFR0.log	100.07	99.24	102.15	99.93	100.28
project-33.doc	79.10	78.73	96.25	82.04	74.39
Propersal.doc	40.44	38.72	74.92	46.99	34.49
Propersal_ส่งหัวเรื่องแล้ว.docx	96.21	94.95	103.77	95.85	95.41
setuppapi.offline.20190318_215229.log	3.69	4.76	63.36	8.64	3.59
setuppapi.offline.log	91.00	90.84	100.18	91.82	91.16
setuppapi.upgrade.log	87.98	87.96	100.04	89.11	88.01
Spect TOR CallCenter.xlsx	82.77	78.29	110.81	79.01	78.19
svnserve.bak	38.96	35.41	115.53	56.15	35.86
WindowsUpdate.log	98.91	72.83	1,001.45	99.28	83.33
การบีบอัดข้อมูลในเครือข่าย(1).pdf	91.90	91.19	99.57	92.73	90.14
การบีบอัดเอกสารเอกซที่เอ็มแอลบนฝั่งเซิร์ฟเวอร์ด้วยการเข้ารหัสแบบฮัฟแมน(1).pdf	91.21	91.49	100.55	92.51	91.40
การพัฒนาอัลกอริทึมการบีบอัดแบบไม่สูญเสียข้อมูลบนเว็บแอปพลิเคชัน_New.docx	93.35	91.08	100.12	92.60	88.88
การศึกษาเปรียบเทียบวิธีบีบอัดข้อมูลที่เหมาะสม - 1.pdf	81.23	81.69	99.26	82.66	79.74
คลังร้อยเอ็ด.txt	22.71	13.07	248.01	15.59	12.84
คำอธิบาย LZMA.pdf	91.22	89.29	99.92	91.85	88.42
คู่มือการใช้งาน SSIS.pdf	95.21	93.13	99.99	95.22	92.43
ตัวอย่างระบบ CallIncome.doc	90.10	88.41	98.83	91.32	85.78
เทคนิคการสืบค้นสารสนเทศและทำเหมืองข้อความ.pdf	91.06	89.71	99.74	91.94	88.47
ใบเสนอราคา.doc	22.17	19.22	51.29	24.27	16.26
สูตรการตั้งสำรอง.xlsx	98.56	98.41	100.42	99.30	98.00
หมวด : Group_Images					
001_55.tif	31.51	27.02	60.65	31.50	24.22
001_60.tif	31.42	26.86	61.52	31.51	23.67
58965.jpg	91.80	92.83	99.39	93.44	92.62
9856.tif	56.99	74.70	98.28	93.52	56.90
add-01.psd	38.19	56.27	88.71	80.28	29.71
BAY090-0001.tif	48.13	65.32	81.81	93.13	47.95
BAY090-0001-0003.tif	48.13	65.32	81.81	93.13	47.95
BAY127.jpg	96.19	99.10	99.50	99.70	98.06
BAY133.png	100.37	100.03	100.06	100.01	100.67

3. ผลการเปรียบเทียบอัตราส่วนการบีบอัดข้อมูล (Compression Ratio) แยกเป็นไฟล์

ชื่อไฟล์	อัลกอริทึม (อัตราส่วน : ร้อยละขนาดไฟล์)				
	BZip2	DEFLATE	Huffman Coding	LZ4	LZMA
BAY133.psd	88.35	93.23	99.63	99.21	85.97
BAY141.gif	100.19	100.03	100.05	100.01	99.44
BAY294.JPG	99.31	99.37	100.08	99.42	100.04
binoculars-01.psd	33.12	57.97	90.47	84.74	25.84
Cancel.gif	169.48	91.66	321.72	90.10	88.18
CD.gif	128.15	68.61	309.26	68.03	63.36
delicious.psd	24.10	30.24	79.20	44.90	19.23
DesktopKeepOnToasting.gif	94.19	95.38	100.76	97.07	91.04
KTB586.JPG	98.92	99.16	100.05	99.31	99.78
KTB638.JPG	99.13	99.24	100.03	99.30	99.89
KTB656.JPG	98.93	99.22	100.06	99.41	99.69
KTB793.JPG	98.59	98.64	100.14	98.78	99.21
logo.png	103.60	98.80	125.49	99.26	99.23
mail.psd	22.99	29.58	77.42	44.00	18.41
mAlNn.psd	65.59	78.63	97.67	95.67	68.32
Musics.png	98.79	97.68	100.42	98.59	94.70
noimage.png	111.48	100.09	144.55	100.26	100.92
Office Publisher.png	99.40	97.35	101.24	98.08	97.80
ok.psd	53.99	71.21	96.16	92.87	58.65
ok.tif	42.64	57.01	82.87	74.18	42.47
OnInstallSnapshot.png	99.49	98.58	99.86	100.01	98.07
oobe-bookend-cortanain.gif	99.22	97.81	101.07	99.48	97.53
Page1_WebChartControl1_ChartsImage.bmp	0.76	1.75	50.75	2.27	0.80
PageNotFound.png	99.57	97.94	106.78	100.04	97.90
Recycle-Bin-Basic-Empty.png	100.58	100.03	100.86	100.01	101.00
reminder.gif	71.80	67.48	100.81	69.49	65.12
s.png	99.68	99.22	100.27	99.72	99.88
sprite.gif	112.26	99.42	147.83	100.29	96.55
TextBox สำหรับ Username.jpg	164.49	86.31	286.12	89.70	83.30
WelcomeFax.tif	65.01	63.38	101.43	66.08	62.19
win7-desktop-green-bg.jpg	100.12	99.17	100.28	99.99	99.64
YaaHoo.psd	44.22	63.42	94.90	90.56	
ZPLDZ.psd	45.76	64.71	95.23	91.48	51.76

3. ผลการเปรียบเทียบอัตราส่วนการบีบอัดข้อมูล (Compression Ratio) แยกเป็นไฟล์

ชื่อไฟล์	อัลกอริทึม (อัตราส่วน : ร้อยละขนาดไฟล์)				
	BZip2	DEFLATE	Huffman Coding	LZ4	LZMA
หมวด : Group_Music					
ก็มาดึคะ - ยุ่งยั้ง กนกนันท์ feat.Night tingle (OFFICIAL MV)(0).mp4	100.18	99.80	100.02	99.91	100.61
ขอบใจเด้อ - ศาล สานศิลป์ เข็ง Music.mp3	97.79	97.87	99.70	98.20	98.06
ขอโอกาสแห่นเด้อ - บอย พนมไพร.mp3	93.94	94.13	98.71	94.39	94.27
คนละฟ้า - แจ็ค ลูกอีสาน (Official Music Video).mp3	96.01	96.13	99.15	96.58	96.27
คืนมาสาหล่า - แจ็คลูกอีสาน (Official MV).mp3	97.33	97.38	99.50	97.74	97.56
คืนมาสาหล่า - แจ็คลูกอีสาน (Official MV).wav	89.85	92.86	96.44	98.34	89.33
แจ๊คลูกอีสาน คำแม่แจ๊ค ลูกอีสาน.mp3	96.69	96.86	99.54	97.26	97.04
เด้อเดียงต่าง เ ล้าโย ไททองคำ.wmv	95.62	95.78	99.24	97.20	95.29
เด้อเดียงต่าง.wmv	90.47	91.60	98.52	93.02	90.77
นอนสาหล่า - สมาร์ท วาทยธรา.mp4	100.31	99.77	100.10	99.90	100.54
นอนสาหล่า.avi	99.77	99.61	100.02	99.82	100.03
นอนสาหล่า.mp4	100.36	99.95	100.03	99.95	101.00
บั้งจ๊กเม็ต.wmv	96.14	96.56	99.35	97.83	95.26
บัดนี้ - แจ๊ค ลูกอีสาน (Official MV).wav	96.92	97.25	97.98	99.67	95.62
ปี่(จน)ป่น - เอ มหาหงษ์ feat.บัว กมลทิพย์ Cover MV น่องน้าฝน พี่ตี๋.mp4	99.91	99.57	100.00	99.66	100.62
เป่ ภาณูชัย พรสวรรค์ ท็อปไลน์EP.1.mp3	99.43	99.49	99.74	99.87	99.69
ผู้สาวมักอ้ายบ - น่องเอเซีย [Cover MV] น่องอินดี้ น่องนุ่น บะเคซิติ้.mp4	100.16	99.77	100.02	99.78	101.02
ฝนเทลงมา - กาเน็ต สะเลอปปี้ [Cover MV] น่องอินดี้ ทีมเด่น บะเคซิติ้.avi	97.49	97.73	99.90	98.76	97.41
ฝนเทลงมา - กาเน็ต สะเลอปปี้ [Cover MV] น่องอินดี้ ทีมเด่น บะเคซิติ้.mp4	100.21	99.82	100.02	99.82	101.08
ฝนเทลงมา - กาเน็ต สะเลอปปี้ [Cover MV] น่องอินดี้ ทีมเด่น บะเคซิติ้.wmv	95.53	94.85	97.56	96.92	94.52
ฝนเทลงมา - กาเน็ต สะเลอปปี้ น่องอินดี้ ทีมเด่น บะเคซิติ้.avi	96.83	97.29	99.82	98.58	96.79
พ้อค้ำน้อย.mp4	100.26	99.86	100.02	99.86	101.13
พ้อค้ำน้อย.wmv	97.98	97.98	99.46	99.10	97.82
ฟังเพลิน3ชม ขับรถ, ทำงาน , บุญบ้าน, เอ อนุชา กุ้ง สุภาพร.mp3	99.64	99.70	99.79	100.01	99.94
ฟังเพลิน3ชม ขับรถ, ทำงาน , บุญบ้าน, เอ อนุชา กุ้ง สุภาพร.wav	82.52	90.78	91.25	100.00	83.56
รถแห่รถยู (ปีเล็งโปงฉิ่ง) - น่อง ทิวเทน Cover MV โดย น่องโปรแกรม(0).mp4	100.31	99.90	100.03	99.93	100.75
รถแห่รถยู (ปีเล็งโปงฉิ่ง) - น่อง ทิวเทน Cover MV โดย น่องโปรแกรม.mp4	100.30	99.89	100.02	99.93	100.75
รถแห่รถยู (ปีเล็งโปงฉิ่ง) - น่อง ทิวเทน Cover MV โดย น่องโปรแกรม.wmv	98.08	97.94	99.49	99.01	97.85
รวมเพลง มนต์แคน แก่นคูน 15 เพลงฮิตจากบ่าวเสียงสุดสะแนน.mp3	99.57	99.65	99.82	99.96	99.86
รวมเพลง มนต์แคน แก่นคูน 15 เพลงฮิตจากบ่าวเสียงสุดสะแนน.wav	96.46	97.89	98.12	99.94	96.29
รวมเพลงลูกทุ่งเจ็บ ๆ โดน ๆ ซึ่ง ๆ ฟังแล้วน้ำตาจิไหล ยอดฮิตใหม่ล่าสุด2019.mp3	98.12	98.24	99.62	98.68	98.42
รวมเพลงฮิต ผู้บ่าวไทบ้าน ซุปตาร์อีสาน เ	99.46	99.76	99.85	100.00	99.86

3. ผลการเปรียบเทียบอัตราส่วนการบีบอัดข้อมูล (Compression Ratio) แยกเป็นไฟล์

ชื่อไฟล์	อัลกอริทึม (อัตราส่วน : ร้อยละขนาดไฟล์)				
	BZip2	DEFLATE	Huffman Coding	LZ4	LZMA
หนุ่ม น.ป.ช. , อส. รอรัก , ดาวมี่ไว้เบิ่ง.mp3					
รักติดไซเรน (My Ambulance) - ไอซ์ พาริส, แพรวา ณิชารัทร [Official MV] Nadao Music.avi	98.37	98.84	99.93	99.55	98.03
หมอล้ำลงทุ่ง ลงนา.mp3	88.00	89.71	96.27	93.00	86.61
อุกกา.avi	99.54	99.53	100.02	99.72	99.67
เอ อนุชา พาย้อนยุค 10 ปีคืนหลัง ฟังลำซิ่ง1990.mp3	95.23	95.72	99.07	97.32	95.21
เอ อนุชา พาย้อนยุค 10 ปีคืนหลัง ฟังลำซิ่ง1990.wav	63.42	86.54	92.43	99.87	60.67
เอาอย่างจ้.mp4	100.27	99.87	100.02	99.88	101.14
Total	81.62	82.30	107.11	87.18	78.52

4. ผลการเปรียบเทียบประสิทธิภาพเวลารวมทุกกระบวนการ (Overall Time) แยกเป็นไฟล์

ชื่อไฟล์	อัลกอริทึม (อัตราส่วน : วินาที)				
	BZip2	DEFLATE	Huffman Coding	LZ4	LZMA
หมวด : Group_Data					
000004.log	2.33	0.24	34.35	0.11	2.02
101_Full.pdf	105.13	6.65	1, 536.24	3.10	119.35
120Page.pdf	52.68	3.79	741.94	1.70	58.94
132079212448625110.pdf	124.34	5.76	2, 013.73	3.04	138.29
2.pdf	5.40	1.49	86.95	0.22	6.14
3.pdf	6.28	1.66	96.39	0.21	6.99
5000568.pdf	104.26	4.94	1, 539.13	3.03	123.13
5000664.pdf	104.36	4.85	1, 512.88	2.40	126.08
5001157.pdf	163.45	7.79	2, 408.76	4.76	178.86
5001797.pdf	357.17	21.78	4, 965.08	12.77	404.74
5100713.pdf	141.72	6.67	1, 995.02	3.53	153.49
5100854.pdf	25.35	6.96	366.01	0.62	28.95
5707319.pdf	33.47	1.61	482.00	0.85	38.01
5707358.pdf	59.65	2.78	847.49	1.41	70.19
5707901.pdf	18.93	1.02	272.95	0.61	21.49
570790189.pdf	11.57	3.22	193.74	0.40	15.52
61080810.pdf	28.81	2.19	468.35	1.08	35.27

4. ผลการเปรียบเทียบประสิทธิภาพเวลารวมทุกกระบวนการ (Overall Time) แยกเป็นไฟล์

ชื่อไฟล์	อัลกอริทึม (อัตราส่วน : วินาที)				
	BZip2	DEFLATE	Huffman Coding	LZ4	LZMA
Append.pdf	10.05	4.56	156.15	2.29	5.61
ASPxPivotGrid.pdf	10.18	2.24	153.87	0.33	11.78
bch01.log	0.09	0.07	0.25	0.07	0.10
Book1.xlsx	83.49	7.88	1, 378.87	4.05	118.26
Book1d.xlsx	13.87	3.31	210.34	0.71	18.87
bootstrap_log.txt	0.06	0.05	0.18	0.06	0.10
CRM.xlsx	305.91	26.15	4, 769.05	15.53	432.81
DataConnections.xlsx	80.72	24.97	1, 329.03	4.10	110.26
DB_EDocuments_backup_2018_03_28_070000_9724092.bak	0.93	0.23	27.71	0.10	2.84
DB_EDocuments_Full_TranserEDoc_Completed.fbak	11.05	2.98	325.84	0.75	31.07
DPINST.LOG	0.86	0.02	0.10	0.02	0.04
h.pdf	6.47	9.19	97.15	0.47	7.64
large_empty.scss.txt	0.77	0.07	396.48	0.08	3.67
ngen.log	0.52	0.09	14.34	0.12	0.83
OKTest.pdf	2.49	0.69	38.91	0.52	2.87
PFRO.log	0.25	0.10	2.73	0.08	0.27
project-33.doc	16.17	3.49	272.90	0.59	19.14
Propersal.doc	0.19	0.08	2.95	0.60	0.30
Propersal_ส่งหัวเรื่องแล้ว.docx	0.17	0.09	1.76	0.07	0.20
setupapi.offline.20190318_215229.log	2.54	0.12	91.55	0.12	3.54
setupapi.offline.log	1.96	1.04	34.11	0.10	1.99
setupapi.upgrade.log	3.99	1.01	67.38	0.16	4.49
Spect TOR CallCenter.xlsx	0.06	0.08	0.31	0.07	0.13
svnservice.bak	0.03	0.01	0.09	0.02	0.04
WindowsUpdate.log	0.02	0.05	0.05	0.20	0.07
การบีบอัดข้อมูลในเครือข่าย(1).pdf	8.63	2.13	132.73	0.36	9.74
การบีบอัดเอกสารเอชทีเอ็มแอลบนฝั่งเซิร์ฟเวอร์ด้วยการเข้ารหัสแบบฮิปแมน(1).pdf	1.11	0.31	17.34	0.28	1.23
การพัฒนาอัลกอริทึมการบีบอัดแบบไม่สูญเสียข้อมูลบนเว็บแอปพลิเคชัน_New.docx	1.71	0.55	29.79	0.43	1.81
การศึกษาเปรียบเทียบวิธีบีบอัดข้อมูลที่เหมาะสม - 1.pdf	0.60	0.25	9.65	0.20	0.69
คลังร้อยเอ็ด.txt	0.02	0.03	0.07	0.01	0.03
คำอธิบาย LZMA.pdf	1.17	0.44	16.64	1.29	1.24

4. ผลการเปรียบเทียบประสิทธิภาพเวลารวมทุกระบวนการ (Overall Time) แยกเป็นไฟล์

ชื่อไฟล์	อัลกอริทึม (อัตราส่วน : วินาที)				
	BZip2	DEFLATE	Huffman Coding	LZ4	LZMA
คู่มือการใช้งาน SSIS.pdf	10.54	2.40	148.51	2.16	11.72
ตัวอย่างระบบ CallIncome.doc	1.73	0.41	26.96	0.22	1.86
เทคนิคการสืบค้นสารสนเทศและทำเหมืองข้อความ.pdf	10.00	2.74	142.12	0.39	11.45
ใบเสนอราคา.doc	0.07	0.05	0.52	0.08	0.16
สูตรการตั้งสำรอง.xlsx	1.42	1.78	20.91	1.47	1.45
หมวด : Group_Images					
001_55.tif	0.69	0.06	0.38	0.07	0.10
001_60.tif	0.10	0.10	0.40	0.06	0.12
58965.jpg	28.13	9.16	293.25	5.74	31.03
9856.tif	154.64	41.75	2, 294.78	58.53	249.25
add-01.psd	1.54	0.39	25.60	0.41	1.62
BAY090-0001.tif	12.04	1.67	125.12	2.27	22.18
BAY090-0001-0003.tif	9.33	1.87	122.00	2.25	19.04
BAY127.jpg	13.23	2.19	151.16	1.77	13.91
BAY133.png	14.27	2.05	153.36	1.97	17.16
BAY133.psd	13.16	6.35	162.54	6.61	14.45
BAY141.gif	13.58	4.29	136.94	1.83	13.84
BAY294.JPG	3.17	0.47	33.54	0.49	3.86
binoculars-01.psd	4.66	1.22	80.77	1.15	4.74
Cancel.gif	0.93	0.08	0.07	0.02	0.07
CD.gif	0.08	0.03	0.06	0.07	0.16
delicious.psd	3.54	0.52	73.79	0.76	4.85
DesktopKeepOnToastimg.gif	0.81	0.41	7.59	0.17	0.78
KTB586.JPG	2.70	0.47	28.64	0.43	2.46
KTB638.JPG	2.69	0.47	29.09	0.48	3.31
KTB656.JPG	3.19	0.58	31.18	0.54	5.42
KTB793.JPG	1.50	0.34	15.62	0.25	1.44
logo.png	0.07	0.09	0.32	0.38	0.11
mail.psd	3.63	0.51	61.72	0.70	4.73
mAlNn.psd	80.72	58.65	989.01	49.08	124.91
Musics.png	1.82	0.34	15.71	0.93	2.55
noimage.png	0.09	0.06	0.24	0.05	0.19
Office Publisher.png	0.67	0.75	5.33	0.14	0.57

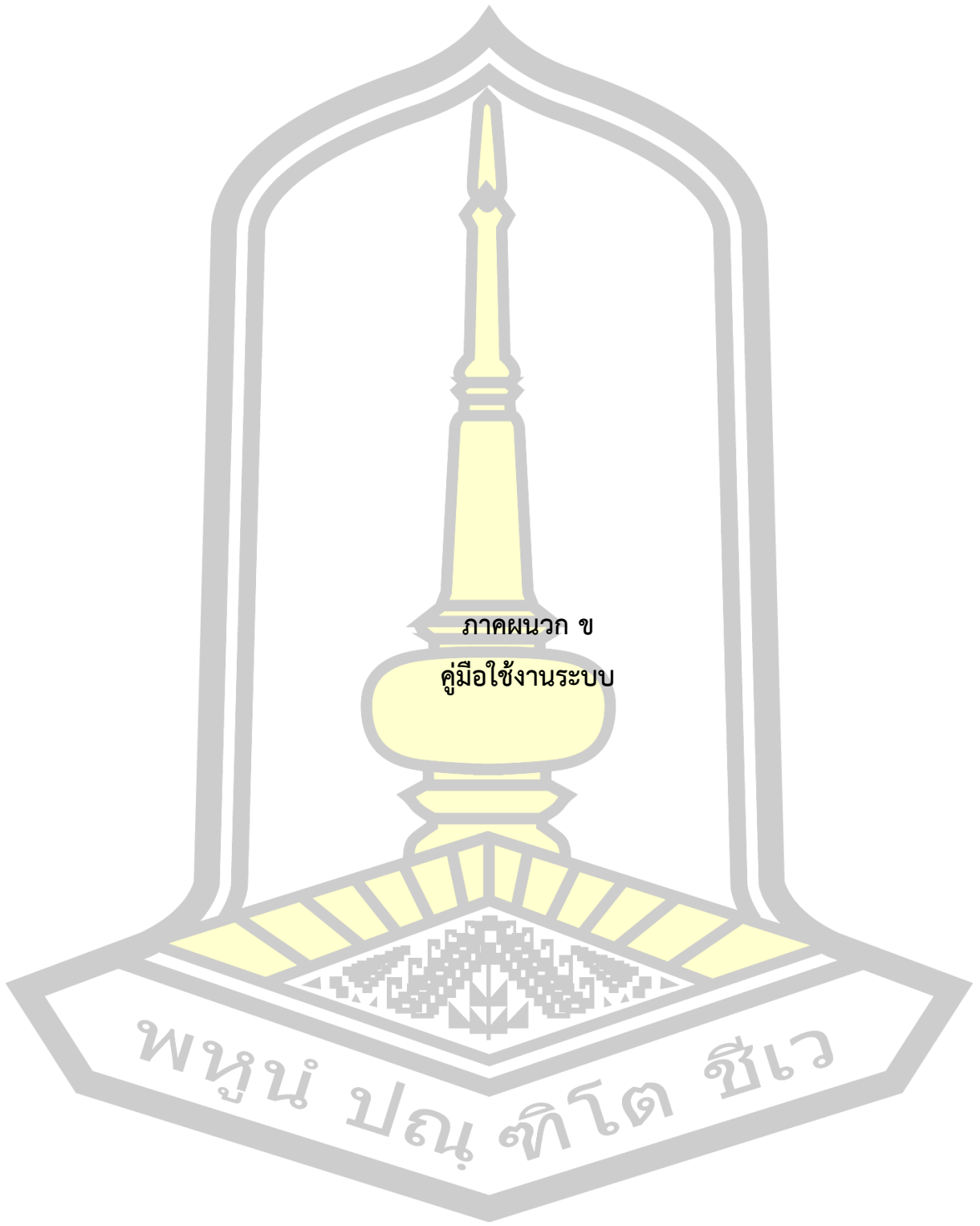
4. ผลการเปรียบเทียบประสิทธิภาพเวลารวมทุกกระบวนการ (Overall Time) แยกเป็นไฟล์

ชื่อไฟล์	อัลกอริทึม (อัตราส่วน : วินาที)				
	BZip2	DEFLATE	Huffman Coding	LZ4	LZMA
ok.psd	183.05	50.96	1, 991.73	117.19	234.05
ok.tif	141.35	64.92	2, 217.33	56.70	236.10
OnInstallSnapshot.png	2.17	0.40	22.68	0.34	1.96
oobe-bookend-cortanain.gif	0.39	0.14	3.42	0.12	0.40
Page1_WebChartControl1_ChartsImage.bmp	1.89	0.08	15.96	0.06	0.61
PageNotFound.png	0.17	0.06	1.08	0.06	0.21
Recycle-Bin-Basic-Empty.png	1.00	0.18	8.43	0.22	1.06
reminder.gif	0.39	0.11	4.20	0.11	0.32
s.png	2.79	0.50	26.76	0.77	3.05
sprite.gif	0.14	0.06	0.19	0.07	0.06
TextBox สำหรับ Username.jpg	0.04	0.05	0.09	0.06	0.09
WelcomeFax.tif	0.24	0.13	2.57	0.09	0.24
win7-desktop-green-bg.jpg	2.55	2.77	25.74	0.38	3.23
YaaHoo.psd	131.85	38.95	1, 853.21	51.96	
ZPLDZ.psd	116.57	49.27	1, 829.60	57.96	264.28
หมวด : Group_Music					
ก็มาดิคะ - ยุ่งยิง กนกนันท์ feat.Night tingle (OFFICIAL MV)(0).mp4	112.34	5.96	2, 108.93	3.33	130.66
ขอบใจเด้อ - ศาล สานศิลป์ เข็ญ Music.mp3	7.47	3.34	121.76	0.25	7.90
ขอโอกาสแทนเด้อ - บอย พนมไพร.mp3	8.02	3.33	124.72	2.02	8.89
คนละฟ้า - แจ็ค ลูกอีสาน (Official Music Video).mp3	9.22	4.76	140.60	0.30	10.33
คืนมาสาหล่า - แจ็คลูกอีสาน (Official MV).mp3	8.06	3.52	130.25	0.23	9.09
คืนมาสาหล่า - แจ็คลูกอีสาน (Official MV).wav	84.64	7.16	1, 544.40	2.84	109.87
แจ๊คลูกอีสาน คำแม่แจ๊ค ลูกอีสาน.mp3	7.51	3.76	129.98	0.22	8.69
เด้อเดียงดาง ลำไย ไททองคำ.wmv	5.88	3.51	104.89	0.26	7.50
เด้อเดียงดาง.wmv	8.41	3.91	145.27	0.34	10.43
นอนสาหล่า - สมาร์ท วาทยิยามา.mp4	6.32	2.68	103.74	0.21	6.97
นอนสาหล่า.avi	87.22	5.32	1, 408.68	2.04	105.07
นอนสาหล่า.mp4	45.34	2.23	697.44	1.13	52.35
บั้งจักเม็ด.wmv	188.49	14.75	2, 953.73	8.49	231.21
บัดนี้ - แจ็ค ลูกอีสาน (Official MV).wav	110.34	8.35	1, 910.72	2.81	131.78
ปี่(จน)ป่น - เอ มหาหิงค์ feat.บัว งามลทิพย์ Cover MV น่องน้ำฝน พี่ดี.mp4	53.40	2.60	788.55	1.35	57.45
เป้ ภาณุชัย พรสวรรค์ ท็อปไลน์EP.1.mp3	131.64	81.12	2, 369.65	3.55	154.97
ผู้สาวมักย้ายบ่ - น่องเอเซีย [Cover MV] น่องอินดี้ น่องนุ่น บะเคซิดี้.mp4	155.53	7.60	150.76	3.95	180.61

4. ผลการเปรียบเทียบประสิทธิภาพเวลารวมทุกกระบวนการ (Overall Time) แยกเป็นไฟล์

ชื่อไฟล์	อัลกอริทึม (อัตราส่วน : วินาที)				
	BZip2	DEFLATE	Huffman Coding	LZ4	LZMA
ฝนเทลงมา - กาเน็ต สะเลอปี [Cover MV] น้องอินดี้ ทีมเต้น บะเคซิติ.avi	33.73	13.56	1, 312.72	1.05	39.73
ฝนเทลงมา - กาเน็ต สะเลอปี [Cover MV] น้องอินดี้ ทีมเต้น บะเคซิติ.mp4	254.28	13.68	133.24	7.47	300.11
ฝนเทลงมา - กาเน็ต สะเลอปี [Cover MV] น้องอินดี้ ทีมเต้น บะเคซิติ.wmv	11.75	11.40	683.13	0.50	14.50
ฝนเทลงมา - กาเน็ต สะเลอปี น้องอินดี้ ทีมเต้น บะเคซิติ.avi	102.84	7.24	1, 852.28	3.27	123.90
พ้อค้ำน้อย.mp4	131.25	6.03	2, 302.13	5.55	149.21
พ้อค้ำน้อย.wmv	69.63	50.12	1, 127.86	2.23	83.07
ฟังเพลิน3ชม ขับรถ, ทำงาน , บุญบ้าน, เอ อนุชา กุ้ง สุภาพร.mp3	307.49	23.84	4, 736.09	8.33	376.95
ฟังเพลิน3ชม ขับรถ, ทำงาน , บุญบ้าน, เอ อนุชา กุ้ง สุภาพร.wav	347.77	31.85	5, 448.09	10.26	476.58
รถแห่รถยู๊ (ปีะโล้งโป้งฉิ่ง) - น้อง ทิวเทน Cover MV โดย น้องโปรแกรม(0).mp4	94.65	4.86	1, 398.21	2.42	107.21
รถแห่รถยู๊ (ปีะโล้งโป้งฉิ่ง) - น้อง ทิวเทน Cover MV โดย น้องโปรแกรม.mp4	117.55	5.71	2, 146.93	3.15	133.13
รถแห่รถยู๊ (ปีะโล้งโป้งฉิ่ง) - น้อง ทิวเทน Cover MV โดย น้องโปรแกรม.wmv	111.45	8.74	2, 014.01	3.82	136.40
รวมเพลง มนต์แคน แก่นคูน 15 เพลงฮิตจากบ่าวเสียงสุดสะแนน.mp3	97.59	69.35	1, 729.12	2.36	116.02
รวมเพลง มนต์แคน แก่นคูน 15 เพลงฮิตจากบ่าวเสียงสุดสะแนน.wav	86.66	6.68	1, 638.44	2.39	105.57
รวมเพลงลูกทุ่งเรื่บ ๆ โคน ๆ ซิ่ง ๆ ฟังแล้วน้ำตาจไหล ยอดฮิตใหม่ล่าสุด2019.mp3	141.30	10.72	2, 657.39	3.80	162.75
รวมเพลงฮิต ผู้บ่าวไทบ้าน ชูปดาร์อีसान l หนุ่ม น.ป.ช. , อส. รอรัก , ตาวมีไว้เบิ่ง.mp3	102.62	7.73	1, 920.81	2.88	119.26
รักติดไซเรน (My Ambulance) - ไอซ์ พาริส, แพรวา นิชาภัทร [Official MV] Nadao Music.avi	230.70	16.44	265.64	7.00	274.61
หมอลำลางทุ่ง ลงนา.mp3	46.08	25.11	742.79	3.24	62.88
อุกก้า.avi	114.92	7.34	2, 199.93	2.93	139.02
เอ อนุชา พาย้อนยุค 10 ปีคืนหลัง ฟังลำซิ่ง1990.mp3	53.70	30.34	884.09	2.49	66.56
เอ อนุชา พาย้อนยุค 10 ปีคืนหลัง ฟังลำซิ่ง1990.wav	101.97	10.39	2, 441.11	3.67	166.46
เอาอย่างงี้.mp4	166.66	7.51	5, 793.30	4.00	188.67
Total	54.87	9.72	837.98	7.38	70.88

พหุบัน ปณ กิโต ชีเว



ภาคผนวก ข
คู่มือใช้งานระบบ

พหุมนุ ปณุ ทิโต ชีเว

1. ขั้นตอนการเข้าใช้งานระบบ

ให้เปิด Internet Explorer หรือ Google Chrome ขึ้นมา จากนั้นระบุ URL :

<http://192.168.1.26/Comperession> ตรงช่อง Addressbar และกดปุ่ม Enter

DEVELOPING LOSSLESS COMPRESSION ALGORITHMS ON WEB APPLICATIONS

การบีบอัดข้อมูล (Data Compression) เป็นการบีบอัด (Compress) เพื่อให้ใช้จำนวนบิตในการจัดเก็บข้อมูลน้อยกว่าเดิม ข้อมูลที่ถูกบีบอัด (Compress) มาแล้วก็ต้องนำมาคลาย (Decompress) หรือถอดรหัสเพื่อให้ได้ข้อมูลเดิมกลับมาก่อนที่จะสามารถนำไปใช้งาน การบีบอัดแบบไม่สูญเสียเป็นที่นิยมในปัจจุบันซึ่งมีทฤษฎีหลายท่านได้พัฒนาอัลกอริทึมเพื่อมาใช้ในการบีบอัดข้อมูล ประกอบไปด้วยอัลกอริทึมดังต่อไปนี้

- Huffman Code
- Deflate
- LZ4
- BZIP2
- LZMA

© 2562 - Data Compression By Seksit

รูปที่ ข1 หน้าหลักโปรแกรม

เมื่อเข้าระบบได้ จะปรากฏหน้าหลักของโปรแกรม ซึ่งประกอบไปด้วยเมนูหลัก ๆ ดังนี้

- Menu Home คือ หน้าหลักของโปรแกรม
- Menu Upload (Compression) คือเมนูในการอัปโหลดไฟล์ลงระบบและบีบอัดไฟล์ข้อมูล
- Menu Automatic Compression คือเมนูในการสร้างต้นแบบการบีบอัดอัตโนมัติ
- Menu File Download (Decompression) คือเมนูในการแสดงผลไฟล์ในระบบและดาวน์โหลดไฟล์ไปใช้งาน
- Menu Jobs Compression คือเมนูในการสร้างการทดสอบการบีบอัดแบบเป็น Jobs


2. ขั้นตอนการอัปโหลดไฟล์เข้าระบบ และทำการบีบอัดไฟล์ข้อมูล

ให้คลิกเลือก -> Menu Upload (Compression) จะปรากฏหน้าที่ทำหน้าที่ในการอัปโหลดไฟล์

ลงระบบ และบีบอัดไฟล์ตามอัลกอริทึมที่เลือก โดยมีหน้าจการทำงาน ดังรูปที่ ข2

Upload File To Compression

อัปโหลดไฟล์จากเครื่องผู้ใช้งานเข้าระบบ เมื่ออัปโหลดไฟล์เสร็จสิ้นระบบจะทำการบีบอัดข้อมูลและบันทึกเก็บที่ Server.



File Group :

- Data
- Images
- Music & Videos **1**

Algorithms :

- Huffman Codes
- BZip2
- DEFLATE **2**
- LZ4
- LZMA

Data File :

3

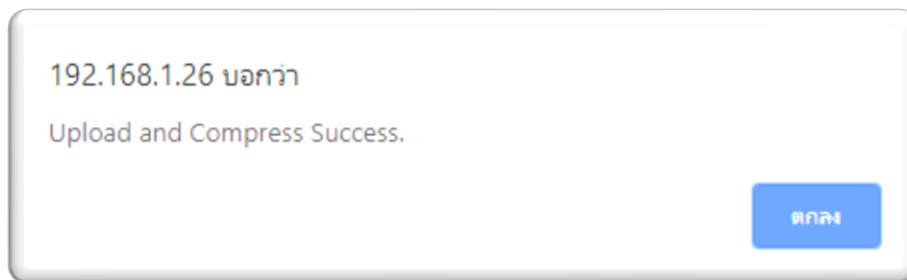
4

© 2562 - Data Compression By Seksit

รูปที่ ข2 หน้าจอการอัปโหลดไฟล์และบีบอัดข้อมูล

หน้าจอในการอัปโหลดไฟล์เพื่อทำการบีบอัดข้อมูล ซึ่งมีขั้นตอนในการใช้งาน ดังนี้

- เลือกหมวดหมู่ไฟล์เอกสารที่ต้องการ
- เลือกอัลกอริทึมที่ต้องการบีบอัด
- เลือกไฟล์ที่ต้องการอัปโหลดเข้าระบบ เมื่อทำการเลือกข้อมูลทุกขั้นตอนแล้ว จากนั้นให้กดปุ่ม **Submit To Server** เพื่ออัปโหลดไฟล์ลงจัดเก็บที่เครื่องบริการเว็บ และทำการประมวลผลการบีบอัดตามอัลกอริทึมที่เลือก เมื่อระบบทำงานเสร็จสิ้นจะปรากฏหน้าต่างแจ้งเตือนการทำงานดังรูปที่ ข3



รูปที่ ข3 แจ้งเตือนสถานการณ์อัปโหลด

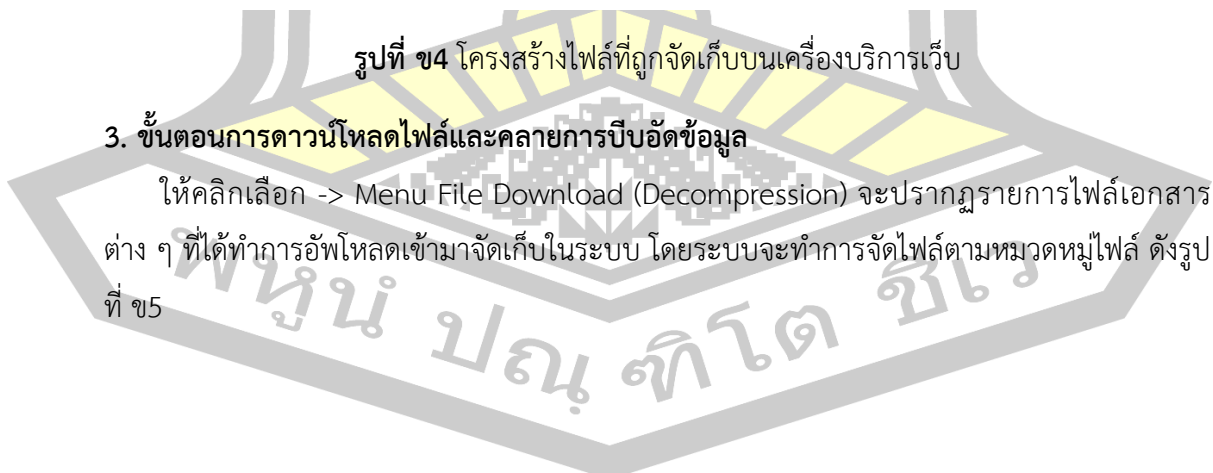
เมื่อไฟล์ถูกประมวลผลการบีบอัดข้อมูลแล้วจะทำการบันทึกจัดเก็บไว้บนเครื่องบริการเว็บ โดยจะทำการเปลี่ยนนามสกุลไฟล์จากเดิมโดยทำการเพิ่มตัวของอัลกอริทึมที่ใช้ในการบีบอัดเข้าไปด้วยทุกไฟล์ ดังรูปที่ ข4

Name	Date modified	Type	Size
การบีบอัดเอกสารเอชทีเอ็มแอลบนฝั่งเซิร์ฟเวอร์ด้วยการเข้ารหัสแบบฮับแมน(1).pdf.hmc	30/10/2562 23:04	HMC File	716 KB
การบีบอัดเอกสารเอชทีเอ็มแอลบนฝั่งเซิร์ฟเวอร์ด้วยการเข้ารหัสแบบฮับแมน(1).pdf.bz2	30/10/2562 23:04	WinRAR archive	650 KB
การบีบอัดเอกสารเอชทีเอ็มแอลบนฝั่งเซิร์ฟเวอร์ด้วยการเข้ารหัสแบบฮับแมน(1).pdf.df	30/10/2562 23:04	DF File	652 KB
การบีบอัดเอกสารเอชทีเอ็มแอลบนฝั่งเซิร์ฟเวอร์ด้วยการเข้ารหัสแบบฮับแมน(1).pdf.lz4	30/10/2562 23:04	LZ4 File	659 KB
การบีบอัดเอกสารเอชทีเอ็มแอลบนฝั่งเซิร์ฟเวอร์ด้วยการเข้ารหัสแบบฮับแมน(1).pdf.lzma	30/10/2562 23:04	LZMA File	651 KB
คำอธิบาย LZMA.pdf.hmc	30/10/2562 23:04	HMC File	690 KB
คำอธิบาย LZMA.pdf.bz2	30/10/2562 23:04	WinRAR archive	630 KB
คำอธิบาย LZMA.pdf.df	30/10/2562 23:04	DF File	616 KB
คำอธิบาย LZMA.pdf.lz4	30/10/2562 23:04	LZ4 File	634 KB
คำอธิบาย LZMA.pdf.lzma	30/10/2562 23:04	LZMA File	610 KB
สูตรการตั้งค่าของ.xlsx.hmc	30/10/2562 23:04	HMC File	733 KB
สูตรการตั้งค่าของ.xlsx.bz2	30/10/2562 23:04	WinRAR archive	722 KB
สูตรการตั้งค่าของ.xlsx.df	30/10/2562 23:04	DF File	721 KB
สูตรการตั้งค่าของ.xlsx.lz4	30/10/2562 23:04	LZ4 File	727 KB
สูตรการตั้งค่าของ.xlsx.lzma	30/10/2562 23:04	LZMA File	719 KB
CompressionHuffman.pdf.bz2	29/1/2563 9:44	WinRAR archive	650 KB

รูปที่ ข4 โครงสร้างไฟล์ที่ถูกจัดเก็บบนเครื่องบริการเว็บ

3. ขั้นตอนการดาวน์โหลดไฟล์และคลายการบีบอัดข้อมูล

ให้คลิกเลือก -> Menu File Download (Decompression) จะปรากฏรายการไฟล์เอกสารต่าง ๆ ที่ได้ทำการอัปโหลดเข้ามาจัดเก็บในระบบ โดยระบบจะทำการจัดไฟล์ตามหมวดหมู่ไฟล์ ดังรูปที่ ข5



Options Search :

File Group :

Data

Images

Music & Videos

File Download :

[Download All](#)

ส่วนแสดงรายชื่อไฟล์

No.	AlgorithmName	TypeOfSize	FileName	OriginalSize	CompressSize	TotalTime	Download
1	LZMA	1KB – 5KB	Cancel.gif	0.00 B	0.00 B	00:00:00:0146	Download
2	LZ4	1KB – 5KB	Cancel.gif	0.00 B	0.00 B	00:00:00:0017	Download
3	DEFLATE	1KB – 5KB	Cancel.gif	0.00 B	0.00 B	00:00:00:0025	Download
4	BZip2	1KB – 5KB	Cancel.gif	0.00 B	0.00 B	00:00:00:0055	Download
5	Huffman Coding	1KB – 5KB	Cancel.gif	0.00 B	0.00 B	00:00:00:0098	Download
6	LZMA	1KB – 5KB	CD.gif	0.00 B	0.00 B	00:00:00:0160	Download
7	LZ4	1KB – 5KB	CD.gif	0.00 B	0.00 B	00:00:00:0054	Download
8	DEFLATE	1KB – 5KB	CD.gif	0.00 B	0.00 B	00:00:00:0055	Download

รูปที่ ข5 แสดงรายชื่อไฟล์ที่อัปโหลดเข้าระบบ

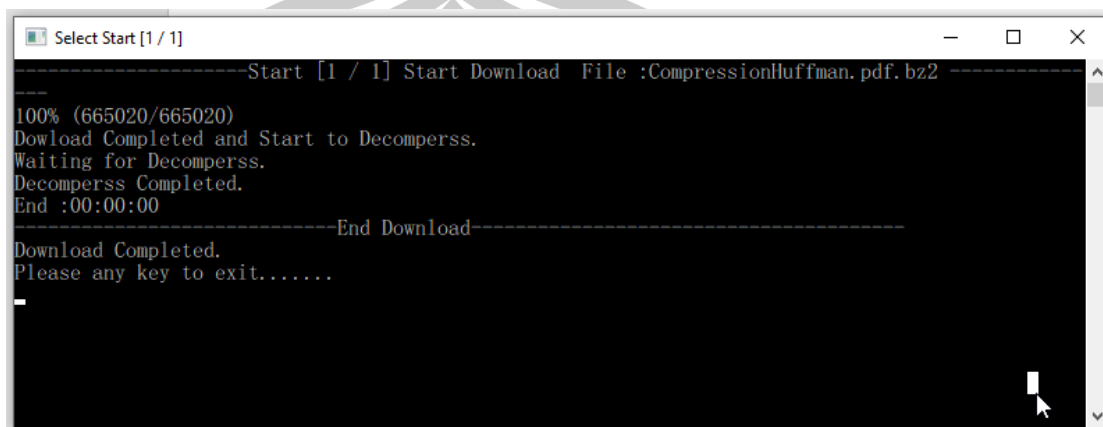
ขั้นตอนในการค้นหาไฟล์ ขั้นตอนแรกต้องกำหนดเงื่อนไขในการค้นหาไฟล์ก่อน เมื่อเลือกหมวดหมู่ไฟล์เสร็จเรียบร้อยแล้วให้กดปุ่ม **Search** ระบบจะทำการแสดงรายการไฟล์ตามที่กำหนดในเงื่อนไข พร้อมทั้งแสดงรายละเอียดในแต่ละไฟล์ดังรูปที่ ข5 ในการดาวน์โหลดไฟล์เอกสารสามารถทำได้โดยคลิกที่ Column Download ที่ไฟล์ที่ต้องการ หลังจากทีคลิกปุ่มดาวน์โหลด จะปรากฏหน้าต่างสอบถามจำนวนรอบที่ต้องการทดสอบ ดังรูปที่ ข6

192.168.1.26 บอกว่า

Number Start for test (x-x):

รูปที่ ข6 แสดงหน้าต่างสอบถามจำนวนรอบในการทดสอบ

หลังจากกดปุ่ม **ตกลง** ระบบจะทำการส่งค่าพารามิเตอร์ไฟล์ที่ต้องการดาวน์โหลดและคลายข้อมูลไปให้โปรแกรมดาวน์โหลดไฟล์ เมื่อโปรแกรมดาวน์โหลดไฟล์เริ่มทำงานจะปรากฏหน้าต่างโปรแกรมดังรูปที่ ข7



รูปที่ ข7 โปรแกรมดาวน์โหลดและคลายการบีบอัดข้อมูล

เมื่อโปรแกรมดาวน์โหลดไฟล์เอกสารเสร็จเรียบร้อยแล้ว โปรแกรมจะทำการคลายการบีบอัดข้อมูล โดยการตรวจสอบนามสกุลไฟล์ก่อนทุกครั้ง เพราะนามสกุลไฟล์จะเป็นตัวบ่งบอกอัลกอริทึมที่ใช้ในการบีบอัด เมื่อโปรแกรมทำการคลายข้อมูลเสร็จเรียบร้อยแล้ว จะแสดงรายละเอียดเวลาการทำงานทั้งหมดและจบการทำงาน

พหุ ประถมศึกษา

4. ขั้นตอนการใช้งานต้นแบบการบีบอัดข้อมูล

ให้คลิกเลือก -> Menu Automatic Compression จะปรากฏหน้าจอการใช้งานระบบการบีบอัดไฟล์อัตโนมัติ ดังรูปที่ ข8

Upload and Auto Compression

อัปโหลดไฟล์จากเครื่องผู้ใช้งานเข้าระบบ เมื่ออัปโหลดไฟล์เสร็จสิ้นระบบจะทำการบีบอัดข้อมูลโดยเลือกวิธีที่ดีที่สุดตามขนาดไฟล์และบันทึกเก็บที่ Server.



File Group :

- Data
- Images 1
- Music & Videos

กฎเกณฑ์การบีบอัด :

- เวลาารวมทุกกระบวนการดีที่สุด 2
- ลดพื้นที่ขนาดไฟล์ดีที่สุด
- กฎทางสถิติที่มีผลสัมฤทธิ์ด้านความเร็ว ดีที่สุด

Data File :

ok.tif 3

4

© 2562 - Data Compression By Seksit

รูปที่ ข8 หน้าหลักต้นแบบการบีบอัดข้อมูล

การใช้งานโปรแกรมต้นแบบการบีบอัดข้อมูลมีขั้นตอนการใช้งานดังนี้

- ขั้นตอนที่ 1 ให้เลือกหมวดไฟล์ที่ต้องการ
- ขั้นตอนที่ 2 ให้เลือกประเภทกฎเกณฑ์ที่ต้องการบีบอัด โดยระบบจะทำการเลือกอัลกอริทึมที่ดีที่สุด ตามกฎเกณฑ์ในแต่ละประเภทกฎเกณฑ์มาให้อัตโนมัติ
- ขั้นตอนที่ 3 ให้เลือกไฟล์ที่ต้องการอัปโหลดเข้าระบบ และทำการบีบอัดไฟล์ข้อมูล
- ขั้นตอนที่ 4 เมื่อเลือกไฟล์ที่ต้องการอัปโหลดเสร็จเรียบร้อยแล้ว ให้กดปุ่ม **Submit To Server** ระบบจะทำการอัปโหลดไฟล์ที่เลือกเข้าระบบก่อน หลังจากนั้นจะทำการตรวจสอบประเภทของไฟล์ที่อัปโหลดเข้ามา พร้อมกับดูขนาดไฟล์ เพื่อนำข้อมูลที่ได้ไปเปรียบเทียบกับผลการทดสอบประสิทธิภาพที่ดีที่สุด เพื่อค้นหาอัลกอริทึมที่เหมาะสมกับประเภทไฟล์นั้น ๆ เมื่อได้อัลกอริทึมที่ดีและเหมาะสมกับไฟล์ ระบบจะทำการประมวลผลการบีบอัดและบันทึกจัดเก็บไฟล์ลงระบบ โดยนามสกุล

ไฟล์ของแต่ละไฟล์จะตามด้วยอัลกอริทึมที่ทำการบีบอัดด้วยเสมอ เนื่องจากไฟล์นามสกุลจะเป็นการบ่งบอกอัลกอริทึมที่ใช้ในการบีบอัดข้อมูล เมื่อมีการคลายข้อมูลระบบจะประมวลผลการคลายข้อมูลได้ถูกต้องตามอัลกอริทึมที่มีการบีบอัด เมื่อโปรแกรมทำงานเสร็จเรียบร้อยจะแสดงหน้าต่างแจ้งเตือนการทำงานดังรูปที่ ข9, ข10, ข11

localhost:5000 says

Upload and Compress Success.

Rule Name : เวลารวมทุกกระบวนการดีที่สุด

Auto Compression : DEFLATE

FileSize :3,804.0000

CompressSize:1,084.0000

Compression Ratio :0.2850

OK

รูปที่ ข9 หน้าต่างแจ้งเตือนการทำงานประเภทเวลารวมดีที่สุด

localhost:5000 says

Upload and Compress Success.

Rule Name : ลดพื้นที่ขนาดไฟล์ดีที่สุด

Auto Compression : LZMA

FileSize :188,604.0000

CompressSize:181,755.0000

Compression Ratio :0.9637

OK

รูปที่ ข10 หน้าต่างแจ้งเตือนการทำงาน ประเภทลดขนาดไฟล์ดีที่สุด

localhost:5000 says

Upload and Compress Success.

Rule Name : กฎทางสถิติที่มีผลสัมฤทธิ์ด้านความเร็ว ดีที่สุด

Auto Compression : LZ4 ->กลุ่มไฟล์ขนาดใหญ่

FileSize :139,394,880.0000

CompressSize:125,177,587.0000

Compression Ratio :0.8980

OK

รูปที่ ข11 หน้าต่างแจ้งเตือนการทำงานประเภทกฎทางสถิติที่มีความน่าจะเป็น ดีที่สุด

ประวัติผู้เขียน

ชื่อ	นายเศกสิทธิ์ พจमार
วันเกิด	20 พฤษภาคม 2528
สถานที่เกิด	66/10 ต.หนองบัวแก้ว อ.พยุหะภูมิพิสัย จ.มหาสารคาม
สถานที่อยู่ปัจจุบัน	212 ม.9 ต.บึงเกลือ อ.เสลภูมิ จ.ร้อยเอ็ด 45120
ตำแหน่งหน้าที่การงาน	นักพัฒนาระบบ ฝ่ายเทคโนโลยีสารสนเทศ
สถานที่ทำงานปัจจุบัน	บริษัทประกันสินเชื่ออุตสาหกรรมขนาดย่อม (บสย.) สำนักงานใหญ่ กรุงเทพมหานคร
ประวัติการศึกษา	2551 ปริญญาวิทยาศาสตรบัณฑิต (วท.บ.) สาขาเทคโนโลยีสารสนเทศ คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม 2563 ปริญญาวิทยาศาสตรมหาบัณฑิต (วท.ม.) สาขาเทคโนโลยีสารสนเทศ คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม

พหุบัณฑิต ชีวะ