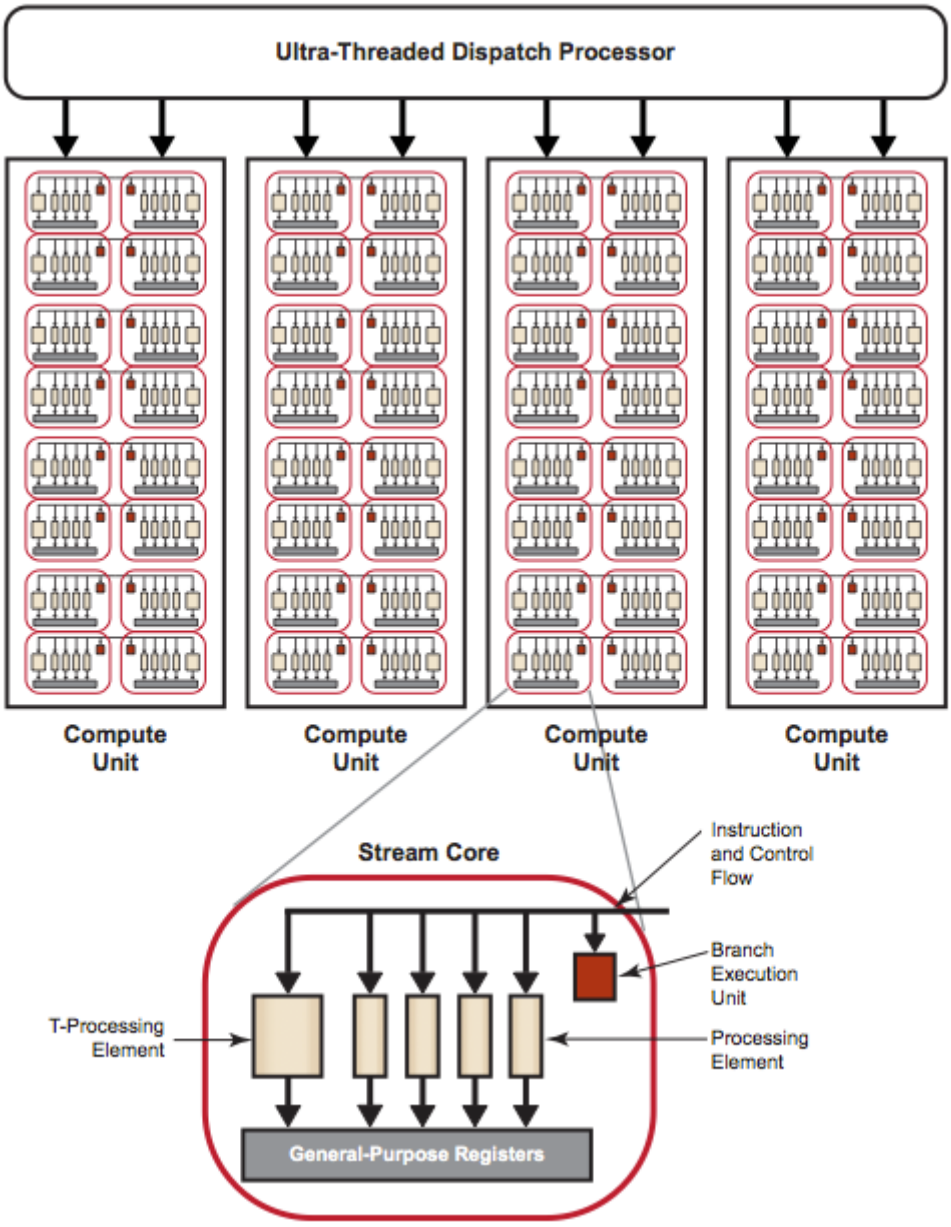
บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 โอเพนซีแอล และเว็บซีแอล

ในปัจจุบันนั้นผู้ผลิตหน่วยประมวลผลได้ออกแบบให้ชิปประมวลผลนั้นมีหลายแกน และได้ใส่หน่วยประมวลผลกราฟฟิกลงไปในหน่วยประมวลผลกลางอีกด้วย หนึ่งในนั้นคือชิปเอพียูซึ่งได้นำเอาหน่วยประมวลผลกราฟฟิกใส่เข้าไปในซีพียูหลายแกนเพื่อลดช่องว่างและเวลาที่ใช้ในการส่งข้อมูลจากหน่วยประมวลผลกลางไปยังหน่วยประมวลผลกราฟฟิก เพื่อให้งานที่ประมวลผลนั้นมีประสิทธิภาพมากยิ่งขึ้น หนึ่งในนั้นคือหน่วยประมวลผลเอพียู ซึ่งมีไดอะแกรมในส่วนของกราฟฟิก ดังภาพที่ 2.1



ภาพที่ 2.1 หน่วยประมวลผลกราฟฟิก
แหล่งที่มา : AMD, 2011

จากภาพที่ 2.1 จะพบว่าภายในจีพียูจะมีหน่วยประมวลผลภายใน ซึ่งภายในหน่วยประมวลผล จะมีแกนประมวลผลสตรีม (Stream Core) ซึ่งมีจำนวนมาก และภายในจะสามารถประมวลผลคำสั่งได้อีก ดังนั้นการประมวลผลแบบขนานโดยใช้จีพียูมีโอกาสที่จะได้ประสิทธิภาพสูงกว่าการประมวลผลโดยการใช้ซีพียู ซึ่งมีจำนวนแกนในการประมวลผลน้อยกว่า ถึงแม้ว่าจะมีความถี่ของสัญญาณนาฬิกา (Clock rate) สูงกว่าก็ตาม

การพัฒนาโปรแกรมในยุคปัจจุบันสามารถนำเอาหน่วยประมวลผลกลาง และหน่วยประมวลผลกราฟฟิกเข้ามาร่วมการทำงานกันได้เพื่อเพิ่มประสิทธิภาพในการประมวลผลแบบขนาน ซึ่งจุดเด่นของหน่วยประมวลผลกราฟฟิกคือมีจำนวนของแกนประมวลผลที่มีมากกว่าแกนประมวลผลกลางซึ่งส่งผลให้สามารถนำมาใช้ในการประมวลผลแบบขนานได้อย่างมีประสิทธิภาพสูงกว่า อย่างไรก็ตามการนำเอาหน่วยประมวลผลกราฟฟิกเข้ามาร่วมการคำนวณด้วย ต้องมีการพัฒนาซอร์สโค้ดโปรแกรมจากการประมวลผลแบบอนุกรม (Sequential) ให้มีการทำงานแบบขนานได้ เพื่อแบ่งงานให้กับหน่วยประมวลผลแต่ละหน่วยได้ทำงานอย่างมีประสิทธิภาพ

ในปัจจุบันนี้มีเฟรมเวิร์ค (Framework) เว็บซีแอล (WebCL) (Khronos, 2010) ในการเขียนโปรแกรมเพื่อที่จะให้โปรแกรมทำงานระหว่างแพลทฟอร์ม (platform) ซีพียู จีพียู ดิจิตัล ซิกนอล โปรเซสเซอร์ (DSPs : Digital Signal Processors) และหน่วยการประมวลผลอื่น ๆ ซึ่งสิ่งนี้จะทำให้สามารถแบ่งงานที่มีการคำนวณขนาดใหญ่ไปให้จีพียู ไม่ว่าจะเป็น NVIDIA หรือ AMD ซึ่งมีปริมาณหน่วยประมวลผลมากกว่าหน่วยประมวลผลกลาง โดยเว็บซีแอลนั้นมีลักษณะการทำงานเช่นเดียวกับโอเพนซีแอล

โอเพนซีแอล (OpenCL : Open Computing Language) (Khronos, 2010; NVIDIA, 2009) เป็นมาตรฐานกลางในการพัฒนาโปรแกรมแบบขนานที่สามารถทำงานได้บนซีพียู (CPU) จีพียู (GPU) และหน่วยประมวลผลอื่น ๆ โดยมาตรฐานนี้นักพัฒนาสามารถนำเอาซอร์สโค้ดไปทำงานได้บนทุกแพลตฟอร์ม (Platform)

โอเพนซีแอลประกอบไปด้วยเอพีไอ (API : Application Program Interface) ที่ใช้ในการประมวลผลแบบขนานระหว่างหน่วยประมวลผลหลายแบบ (heterogeneous processors) และข้ามแพลตฟอร์ม โดยใช้มาตรฐานที่รู้จักกันดีคือ

1) รองรับทั้งการคำนวณข้อมูล และงานแบบขนาน (data parallelism, task parallelism)
2) นำเอาภาษา C มาใช้ในการพัฒนาให้มีการคำนวณแบบขนาน
3) มีการกำหนดค่าตัวเลขและการคำนวณโดยยึดหลักตามมาตรฐาน IEEE754
4) มีการกำหนดการตั้งค่าต่าง ๆ สำหรับอุปกรณ์มือถือและสมองกลฝังตัว
5) สามารถนำเอามาประยุกต์กับ OpenGL OpenGL ES และ API อื่น ๆ ที่เกี่ยวข้องกับงานด้านกราฟฟิก ได้อย่างมีประสิทธิภาพ

ในการอธิบายโอเพนซีแอลนั้นสามารถอธิบายได้โดยใช้รูปแบบได้สี่รูปแบบ คือ โมเดลแพลฟอร์ม(Platform Model) โมเดลหน่วยความจำ(Memory Model) โมเดลการประมวลผล (Execution Model) และ โมเดลการเขียนโปรแกรม(Programming Model)

โมเดลแพลตฟอร์ม ประกอบไปด้วยโฮสต์ (Host) และอุปกรณ์โอเพนซีแอล(OpenCL devices) โดยอุปกรณ์โอเพนซีแอลจะถูกแบ่งออกเป็น 1 หรือมากกว่า 1 หน่วยการคำนวณ (Compute Units : CUs) ซึ่งแต่ละหน่วยประมวลผลจะถูกแบ่งออกเป็น 1 หรือมากกว่า 1 หน่วยประมวลผล (Processing Elements : PEs) โดยการคำนวณต่าง ๆ บนอุปกรณ์จะคำนวณในระดับหน่วยประมวลผล

ในการพัฒนาโปรแกรมด้วยโอเพนซีแอลนั้นจะต้องพัฒนาทั้งฝั่งโฮสต์ และฝั่งอุปกรณ์ ซึ่งฝั่งโฮสต์นั้นจะพัฒนาโปรแกรมปกติ และโค้ดส่วนการจัดการโอเพนซีแอล ไม่ว่าจะเป็นการจองพื้นที่หน่วยความจำ การส่งข้อมูล ในส่วนของโค้ดบนอุปกรณ์จะเกี่ยวข้องกับคำสั่งการประมวลผล ที่ต้องการให้หน่วยประมวลผลในอุปกรณ์ทำงาน

โมเดลการประมวลผลเป็นการอธิบายระหว่าง เคอร์นอล (kernels) ที่ทำงานบน 1 หรือมากกว่า บนอุปกรณ์โอเพนซีแอล และโปรแกรมบนโฮสต์ (host programs) โดยจะมีการแบ่งออกเป็นงานซึ่งงาน 1 งานจะถูกเรียกว่า work-item ซึ่งจะทำงานอยู่ภายในกลุ่มหรือ work-group

ในการจัดการของเคอร์นอลต้องมีการจัดการทรัพยากรดังนี้

1) อุปกรณ์
2) เคอร์นอล
3) โปรแกรม
4) หน่วยความจำ

ในส่วนของโฮสต์โปรแกรมจะมีการสร้างคอนเทกซ์ (context) โดยจะมีการทำงานควบคู่ไปกับอุปกรณ์ผ่านคิวคำสั่ง (command-queue) โดยแต่ละคิวคำสั่งจะมีการโต้ตอบกับแต่ละอุปกรณ์ ซึ่งจะมีการจัดการคำสั่งสามแบบ ดังนี้

1) คำสั่งเคอร์นัล
2) คำสั่งหน่วยความจำ
3) คำสั่งซิงโครไนเซชัน

โมเดลการประมวลผล เป็นการนำเอาตัวชี้ที่เรียกว่า NDRange ไปแมพลงบนอุปกรณ์เพื่อให้สามารถจัดงานให้เหมาะสมกับอุปกรณ์ที่มีอยู่อย่างจำกัดได้อย่างมีประสิทธิภาพ ซึ่งในโมเดลนี้จะมีวัตถุเคอร์นอล ที่ถูกสร้างขึ้นโดยจะมีการกำหนดค่าอาร์กิวเมนท์ โฮสต์โปรแกรมจะนำเข้าวัตถุเคอร์นอลไปยังคิวคำสั่ง และทำงานบนอุปกรณ์ เมื่อทำงานเสร็จ รับค่าส่งกลับมายังโฮสต์

ในการทำงานของโมเดลนี้จะมีส่วนของเคอร์นอลที่จะรับค่าเข้าสู่อุปกรณ์ ตามลำดับของคิวคำสั่งและทำงานตามลำดับเมื่อทำงานเสร็จเรียบร้อยจะทำการซิงโครไนเซชันเพื่อรอทำคำสั่งถัดไปเมื่อทำงานแบบขนานเสร็จเรียบร้อยเพื่อให้คำตอบในการคำนวณถูกต้อง เนื่องจากการประมวลผลแบบขนานต้องมีการรวมข้อมูลก่อนไปคำนวณในส่วนที่อนุกรมถัดไป

โมเดลหน่วยความจำ ในส่วนนี้จะอธิบายเรื่องโครงสร้างของหน่วยความจำ ของโอเพนซีแอล ขณะที่โปรแกรมกำลังทำงาน คอนเทกซ์จะทำงานโดยจะมีหลายอุปกรณ์ทำงานร่วมกันภายในคอนเทกซ์ได้ ดังนั้นภายในคอนเทกซ์นั้นจะสามารถเข้าถึงหน่วยความจำของทุกอุปกรณ์ได้เช่นกัน ภายในอุปกรณ์ 1 ตัวจะมีหลายกลุ่มงานที่จะทำงานไปพร้อม ๆ กันแบบขนาน และภายในกลุ่มงาน 1 กลุ่มจะมีหลายหน่วยงานที่ทำงานพร้อม ๆ กัน ซึ่งส่วนประกอบของโมเดลหน่วยความจำจะถูกแบ่งออกเป็น 2 ส่วน

1) หน่วยความจำโฮสต์ เป็นหน่วยความจำที่โฮสต์ใช้งานได้ โดยจะเป็นหน่วยความจำที่อยู่นอกการควบคุมของโอเพนซีแอล มีหน้าที่เก็บข้อมูลก่อนส่งเข้าสู่โอเพนซีแอล

2) หน่วยความจำอุปกรณ์ เป็นหน่วยความจำที่เคอร์นอลสามารถใช้งานได้และทำงานอยู่บนอุปกรณ์โอเพนซีแอล

หน่วยความจำอุปกรณ์สามารถจำแนกได้อีก 4 ประเภท คือ

1) หน่วยความจำโกลบอล เป็นส่วนที่อนุญาตให้เขียนและอ่านได้จากทุก ๆ หน่วยงานในทุกกลุ่มงาน
2) หน่วยความจำคงที่ เป็นส่วนของหน่วยความจำโกลบอลที่อนุญาตให้เพียงแต่อ่านเท่านั้น โดยจะถูกจองพื้นที่ตั้งแต่การทำงานฝั่งโฮสต์
3) หน่วยความจำโลคอล เป็นส่วนของหน่วยความจำที่อ่านได้แต่ภายในกลุ่มงานเท่านั้น กลุ่มงานอื่นไม่สามารถเข้าถึงได้
4) หน่วยความจำไพรเวท เป็นส่วนของหน่วยความจำที่สามารถอ่านเขียนได้เฉพาะหน่วยงานเท่านั้น หน่วยงานอื่น ๆ ไม่สามารถเข้าถึงได้

ในแต่ละระดับนั้นจะมีความเร็วในการเข้าถึงแตกต่างกัน หน่วยความจำไพรเวทจะมีความเร็วที่สูงกว่า หน่วยความจำโลคอล และโกลบอล ตามลำดับ ซึ่งนักพัฒนาจะต้องมีการจัดการหน่วยความจำที่ดี เพื่อให้ความเร็วในการทำงานของโปรแกรมที่ถูกพัฒนาด้วยโอเพนซีแอลมีประสิทธิภาพสูงยิ่งขึ้น

เว็บซีแอลนั้นจะมีการทำงานที่แตกต่างจากโอเพนซีแอลเล็กน้อย โดยจะแบ่งออกเป็น 3 ส่วนหลัก ดังนี้

1) ส่วนของจาวาสคริปต์ ที่มีหน้าที่จัดการงานในโฮสต์เท่านั้น ยังไม่ได้ติดต่อกับอุปกรณ์โดยตรง ซึ่งส่วนนี้มีไว้ใช้ในการจองพื้นที่ ไม่ว่าจะเป็นการจองพื้นที่รูป ในฝั่งโฮสต์ หรือพื้นที่ส่วนอื่น

2) ส่วนของเว็บบราวเซอร์ ในส่วนนี้จะเป็นส่วนที่จาวาสคริปต์ติดต่อกับส่วนเสริมเว็บซีแอล โดยส่วนนี้จะเป็นคำสั่งที่เกี่ยวกับการจองพื้นที่ที่จำเป็นในการพัฒนาโปรแกรมด้วยเว็บซีแอลก่อนส่งไปให้กับอุปกรณ์อื่น

3) ส่วนของโอเพนซีแอล ส่วนนี้จะทำงานร่วมกับอุปกรณ์เป็นส่วนที่จัดการอุปกรณ์ไม่ว่าจะเป็นหน่วยประมวลผลกลาง หน่วยประมวลผลกราฟิก เพื่อให้คำนวณและประมวลผลบนอุปกรณ์ได้อย่างถูกต้อง

ขั้นตอนในการทำงานของเว็บซีแอลแบ่งออกเป็น 2 ส่วนคือ ส่วนการเตรียมอุปกรณ์ให้พร้อมก่อนการทำงาน และการดำเนินงานของเคอร์นอล

การเตรียมอุปกรณ์เป็นขั้นตอนที่ทำให้อุปกรณ์พร้อมก่อนการทำงานโดย มีขั้นตอนดังนี้

1) ตรวจสอบว่าเว็บบราวเซอร์สามารถทำงานร่วมกับเว็บซีแอลได้
2) ตรวจสอบแพลตฟอร์ม
3) สร้างคอนเทกซ์
4) สร้างโปรแกรม และคอมไพล์โปรแกรม
5) สร้างเคอร์นอล

การดำเนินงานของเคอร์นอล จะเกิดขึ้นหลังจากสร้างเคอร์นอลแล้วโดยขั้นตอนนี้ จะมีการจัดการกับบัฟเฟอร์ ให้กับอุปกรณ์ การกำหนดตัวแปรเพื่อเชื่อมต่อเข้ากับเคอร์นอลเพื่อให้สามารถส่งข้อมูลเข้าไปยังเคอร์นอลได้ โดยมีขั้นตอนดังนี้

1) การสร้างบัฟเฟอร์
2) การตั้งค่าอาร์กิวเมนต์
3) การสร้างคิวคำสั่ง
4) การเขียนค่าลงในบัฟเฟอร์ของอุปกรณ์โอเพนซีแอล
5) การรันเคอร์นอล
6) การอ่านผลลัพธ์ออกจากอุปกรณ์โอเพนซีแอล

เมื่อทำขั้นตอนดังกล่าวเสร็จแล้วต้องมีการลบบัฟเฟอร์ออกเพื่อให้พื้นที่ในหน่วยความจำอุปกรณ์สามารถนำกลับมาใช้ใหม่ได้ในการทำงานครั้งถัดไป

2.2 เอชทีเอ็มแอล 5

เอชทีเอ็มแอล 5 เป็นมาตรฐานใหม่ของภาษาเอชทีเอ็มแอล (Hyper Text Markup Language : HTML) เป็นภาษาที่ใช้ในการแสดงข้อมูลบนหน้าเว็บ โดยภาษาเอชทีเอ็มแอล 5 เพิ่มความสามารถให้มากขึ้นให้นักพัฒนาสามารถพัฒนาเว็บได้อย่างสะดวกและมีประสิทธิภาพมากยิ่งขึ้น

ความสามารถของเอชทีเอ็มแอล 5 ที่เพิ่มขึ้นมาเช่น ซีแมนทิค(Semantics) เป็นไวยากรณ์ของภาษาที่เพิ่มขึ้นมาในการจัดการแอทริบิว (Attribute) ทำให้เว็บบราวเซอร์เข้าใจความหมายของวัตถุแต่ละชิ้นมากขึ้น ออฟไลน์และสตอเรจ (Offline & Storage) ซึ่งทำให้สามารถใช้งานเว็บแบบออฟไลน์ได้ และยังสามารถเก็บข้อมูลไว้บนเครื่องของผู้ใช้งานได้ นอกจากนี้ยังมีความสามารถในการเข้าถึงอุปกรณ์เช่น จีโอโลเคชัน (Geolocation API) เพื่อเข้าถึงพิกัดของอุปกรณ์ของผู้ใช้ การเข้าถึงกล้องถ่ายภาพของอุปกรณ์ ไมโครโฟน ไจโรสโคป (Gyroscope) สมุดที่อยู่ (Address book) ซึ่งสามารถเรียกใช้ได้จากภาษาจาวาสคริปต์ (Javascript) ความสามารถในการติดต่อกับเซิร์ฟเวอร์โดยมีการติดต่อผ่านซ็อกเก็ตเป็นแบบสองทาง และการส่งข้อมูลทางเดียวจากเซิร์ฟเวอร์ไปยังเครื่องลูกข่าย มัลติมีเดีย (Multimedia) เป็นความสามารถใหม่ของภาษานี้โดยภาษาเอชทีเอ็มแอลรุ่นก่อนหน้าต้องมีการติดตั้งปลั๊กอิน (Plugins) เพื่อใช้ในการเล่นมัลติมีเดียบนเว็บบราวเซอร์ แต่เมื่อใช้ภาษาเอชทีเอ็มแอล 5 ผู้ใช้ไม่จำเป็นต้องติดตั้งปลั๊กอินอีก การรองรับกราฟิกและเอฟเฟก 3 มิติ ทำให้ผู้ใช้สามารถใช้งานกราฟิกแบบเวกเตอร์ (Scalable Vector Graphics : SVG) ใช้งานแคนวาส (Canvas) ซึ่งใช้ในการวาดภาพลงไปในหน้าเว็บ เว็บจีแอล (WebGL) ในการวาดภาพ 3 มิติ ได้

ในงานวิจัยนี้จะนำเอาเอชทีเอ็มแอล 5 มาใช้ในการสร้างเว็บในการตกแต่งภาพร่วมกันกับภาษาจาวาสคริปต์ เพื่อให้สามารถแสดงผลโปรแกรมบนหน้าเว็บได้

2.3 จาวาสคริปต์

จาวาสคริปต์เป็นภาษาโปรแกรมซึ่งมีการแปลความหมายแบบทีละคำสั่งโดยใช้อินเทอพรีตเตอร์ (Interpreter) ในการแปลความหมาย ซึ่งได้รับการพัฒนาขึ้นโดย เน็ตสเคป (Netscape) เพื่อให้เว็บเพจสามารถแสดงเนื้อหาที่มีการเปลี่ยนแปลงตามเงื่อนไข หรือการเปลี่ยนแปลงตามสภาพแวดล้อม และตอบโต้กับผู้ใช้ได้มากขึ้น

จาวาสคริปต์เป็นภาษาโปรแกรมเชิงวัตถุ (Object-Oriented Programming) ซึ่งทำให้สะดวกต่อการพัฒนาโปรแกรม สามารถทำงานข้ามแพลตฟอร์ม (Platform) ได้ร่วมกับภาษาเอชทีเอ็มแอล ผ่านเว็บบราวเซอร์

ภาษาจาวาสคริปต์มีความสามารถดังนี้

1) สามารถพัฒนาโปรแกรมได้ง่าย โดยการพัฒนาร่วมกับภาษาเอชทีเอ็มแอล
2) มีคำสั่งที่สามารถตอบโต้กับผู้ใช้ ทำให้สะดวกในการพัฒนาโปรแกรมในส่วนติดต่อกับผู้ใช้ เช่นมี กล่องเพื่อทำเครื่องหมาย กล่องข้อความ ทำให้สามารถพัฒนาโปรแกรมเพื่อใช้ในการติดต่อกับผู้ใช้ได้ง่ายขึ้น
3) สามารถพัฒนาโดยการเปลี่ยนแปลงค่าของภาษาเอชทีเอ็มแอลได้ (HTML Elements) ทำให้สามารถพัฒนาโปรแกรมเพื่อเปลี่ยนรูปแบบของหน้าเว็บเพจได้
4) สามารถใช้ในการตรวจสอบข้อมูล เช่นการกรอกข้อมูลในเว็บไซต์ เช่นอีเมล หากกรอกข้อมูลผิดสามารถแสดงป๊อปอัป (Pop up) เพื่อแสดงผลว่ามีการกรอกผิดพลาด
5) สามารถสร้างคุกกี้ (Cookies) ใช้ในการเก็บข้อมูลของผู้ใช้คอมพิวเตอร์ได้

ในงานวิจัยนี้จะใช้ภาษาจาวาสคริปต์ในการพัฒนาโปรแกรมเพื่อตอบโต้กับผู้ใช้งาน และสั่งคำสั่งของเว็บซีแอลในฝั่งโฮสต์ เพื่อเตรียมข้อมูลต่าง ๆ ก่อนส่งข้อมูลไปยังฝั่งอุปกรณ์ของเว็บซีแอล

2.4 การประมวลผลภาพ

การประมวลผลภาพ คือการนำภาพมาประมวลผลหรือคิดคำนวณด้วยคอมพิวเตอร์ เพื่อให้ได้ข้อมูลภาพที่ต้องการ เช่น การทำให้ภาพมีความคมชัดมากขึ้น การจัดการสัญญาณรบกวน การแบ่งส่วนของวัตถุที่สนใจ ซึ่งกระบวนการเหล่านี้ทำให้ผู้ใช้สามารถนำภาพไปวิเคราะห์และนำไปใช้ประโยชน์ที่ต้องการ เช่น ระบบรู้จำลายนิ้วมือ การสแกนม่านตา การคัดแยกจดหมายของระบบไปรษณีย์ การเก็บข้อมูลรถเข้าออกในอาคารโดยการตรวจสอบป้ายทะเบียนรถ การใช้ระบบรู้จำใบหน้าของบุคคลในการตรวจสอบคนเข้าเมือง ซึ่งระบบเหล่านี้ใช้การประมวลผลภาพ และต้องมีการทำซ้ำ ซึ่งหากใช้มนุษย์ในการวิเคราะห์จะทำให้ใช้เวลามากกว่าการใช้คอมพิวเตอร์ และสิ้นเปลืองแรงงาน ทำให้เกิดความล้าระหว่างการทำงานและเกิดความผิดพลาดในที่สุด ซึ่งคอมพิวเตอร์จึงเข้ามามีบทบาทในการทำงานแทนมนุษย์

การประมวลผลภาพมี 2 รูปแบบคือ แบบราสเตอร์ (Raster) หรือบิตแมป (Bitmap) โดยภาพเหล่านี้ประกอบขึ้นจากจุดสีต่าง ๆ ที่มีจำนวนคงที่ โดยจะมีการประมวลผลแบบอาศัยการอ่านค่าสีในแต่ละพิกเซล ซึ่งเรียกอีกอย่างว่าบิตแมป โดยจะเก็บค่าข้อมูลเป็น 0 และ 1 และในแต่ละพิกเซลจะมีการเก็บค่าสีที่เจาะจงในแต่ละตำแหน่ง โดยภาพประเภทนี้ มีข้อดีคือ เหมาะสำหรับภาพที่ต้องการระบายสี สร้างสี หรือกำหนดสีที่ต้องละเอียดและสวยงามได้ง่าย ซึ่งปัญหาของภาพลักษณะนี้คือ เมื่อมีการขยายภาพจะทำให้ภาพมีขนาดใหญ่ และเปลืองพื้นที่ ตัวอย่างของไฟล์ประเภทนี้คือ .BMP .PCX .TIF .JPG .GIF .MSP .PNG .PCT เป็นต้น

การประมวลผลแบบเวกเตอร์ เป็นภาพที่ประกอบขึ้นด้วยจุดสีต่าง ๆ โดยคอมพิวเตอร์จะสร้างให้แต่ละส่วนของภาพเป็นอิสระต่อกัน โดยแยกชิ้นส่วนของภาพทั้งหมดออกเป็นเส้นตรง รูปทรงส่วนโค้ง โดยใช้ความสัมพันธ์ทางคณิตศาสตร์ ซึ่งมีสีและตำแหน่งสีที่แน่นอน ดังนั้นหากมีการเปลี่ยนแปลงขนาดภาพทำให้ความละเอียดภาพไม่เปลี่ยนแปลง

การประมวลผลภาพมีรูปแบบต่าง ๆ เช่น เบลอ (Blur) เป็นการลดความละเอียดของภาพลงตามส่วนที่ต้องการ การเปลี่ยนแปลงความสว่าง (Brightness) เป็นการปรับความสว่าง คอนทราสต์ (Contrast) เป็นการเปลี่ยนแปลงความแตกต่างระหว่างโทนมืดและโทนสว่าง การตัดภาพ (Crop) เป็นการตัดบางส่วนของภาพตามที่ต้องการ การเปลี่ยนภาพสี เป็นขาวดำ (Black & White) การกลับสี (Invert Color) เป็นการกลับสีที่อยู่ตรงข้ามกับสีเดิม การเปลี่ยนแปลงสัญญาณรบกวน (Noise) สัญญาณรบกวนนั้นเป็นสิ่งที่เกิดขึ้นกับภาพซึ่งทำให้ภาพหยาบ ซึ่งในการถ่ายภาพมักเกิดจากการเปลี่ยนแปลงค่า ISO ให้เพิ่มมากขึ้นเพื่อให้สามารถรับภาพที่มีความสว่างน้อยได้ จะทำให้เกิดสัญญาณรบกวนมากขึ้น ซึ่งทำให้ภาพไม่สวยงาม การหมุนภาพ (Rotate) เป็นการหมุนภาพไปในแนวที่ต่างกัน

2.5 งานวิจัยที่เกี่ยวข้อง

เว็บบราวเซอร์สมัยใหม่ส่วนใหญ่จะสามารถสร้างแอปพลิเคชัน โดยใช้ภาษาจาวาสคริปต์ในการพัฒนา โดยสามารถใช้เว็บซีแอล ในการเขียนโปรแกรมโดยให้ทำงานบนซีพียู และจีพียู ในการทำงาน อย่างไรก็ตามในการพัฒนาโปรแกรมปกติสามารถทำงานได้เพียง 1 อุปกรณ์ โดยงานวิจัยของ Xianglan Piao, et al [4]. ได้ทดสอบเว็บซีแอลด้วยสี่โปรแกรม ซึ่งการใช้เฟรมเวิร์คนี้สามารถเพิ่มความเร็วได้เพิ่มขึ้น 65% เมื่อเทียบกับการทำงานด้วยจาวาสคริปต์ปกติ และเพิ่มขึ้น 33% เมื่อเปรียบเทียบกับการทำงานบนจีพียูอย่างเดียว

งานวิจัย CrowdCL ของ Tommy MacWilliam, et al [5]. ได้นำเสนอเฟรมเวิร์คที่ใช้ในการพัฒนาแอปพลิเคชันบนเว็บโดยใช้โอเพนซีแอลที่ทำงานบนจีพียู โดยในการทำงานนั้นมีประสิทธิภาพมากยิ่งขึ้นเมื่อเทียบกับการทำงานบนจาวาสคริปต์ โดยที่ผลลัพธ์ปัญหาทอมสัน ซึ่งผลลัพธ์ที่ได้ CrowedCL นั้นทำงานได้อย่างมีประสิทธิภาพเมื่อเทียบกับการทำงานโดยใช้แกนเดียว
ในการเขียนโปรแกรมบนเว็บนั้นจำเป็นจะต้องใช้ภาษาจาวาสคริปต์ซึ่งเป็นภาษาที่ทำงานได้บนหน่วยประมวลผลเพียง 1 แกน ซึ่งในงานวิจัยการพัฒนาโปรแกรมแบบขนานบนเว็บของ Stephen Herhut

et al[6]. ได้มีการทำคอมไพเลอร์โพรโทไทป์แบบ JIT ในบราวเซอร์ไฟร์ฟอกซ์ (Firefox) โดยมีการทดสอบกับการสร้างภาพสามมิติ ซึ่งผลลัพธ์ที่ได้เมื่อเพิ่มงานที่ทำแบบขนานให้มากขึ้น เฟรมต่อวินาทีของการทำงานแบบขนานจะมากยิ่งขึ้นตามไปด้วย

นอกจากนี้ยังมีงานวิจัยเว็บซีแอลสำหรับการเพิ่มความเร็วเว็บแอปพลิเคชันบนฮาร์ดแวร์โดย Won Jeon et al.[7] ซึ่งในงานวิจัยชิ้นนี้ได้มีการนำเอาภาษาเอชทีเอ็มแอลรุ่น 5 (HTML5) ซึ่งเป็นเอชทีเอ็มแอลรุ่นใหม่ มาทำงานร่วมกับภาษาจาวาสคริปต์ โดยมีการนำเอาเว็บแอปพลิเคชันมาพัฒนาร่วมกันกับการประมวลผลร่วมระหว่างซีพียู จีพียู โดยทดสอบร่วมกับ Sobel Filter N-body simulation และ Deformable body simulation ซึ่งทั้ง 3 อัลกอริธึมที่นำมาทดสอบนั้นใช้พลังในการประมวลผลสูงมาก โดยความเร็วที่สูงขึ้นของ Sobel filter ขึ้นถึง 13 เท่า ที่ภาพขนาด 256 x 256 พิกเซล N Body simulation ที่ 1024 พาติเคิล มีความเร็วเพิ่มขึ้นสูงถึง 23 เท่า และ Deformable body simulation มีความเร็วสูงสุดถึง 116 เท่า

งานวิจัยการใช้จาวาสคริปต์และเว็บซีแอลสำหรับการคำนวณทางคณิตศาสตร์ ของ Faiz Khan et al[8]. ได้มีการเปรียบเทียบประสิทธิภาพของการทำงานโดยใช้ภาษาจาวาสคริปต์ เทียบกับการนำเอาเว็บซีแอลมาพัฒนาแบบขนาน โดยใช้อัลกอริธึมของ Dense Linear Algebra, Sparse Linear Algebra, Spectral Methods, N-Body, Structured grids, Unstructed grids, Map Reduce, Combinatorial logic, Graph traversal, Dynamic programming, Backtrack and Branch-and-bound และ Graphical models ซึ่งผลลัพธ์ที่ได้พบว่าอัลกอริธึมส่วนใหญ่จะทำงานได้ดีกว่าการทำงานบนจาวาสคริปต์ปกติเพียงอย่างเดียว นอกจากนี้ยังพบว่าเวลาที่ใช้ส่วนใหญ่สูญเสียไปกับการตั้งค่าฟังก์ชันก่อนการใช้เว็บซีแอล แต่อย่างไรก็ตามการคำนวณแบบขนานส่วนใหญ่ยังใช้เวลาในการทำงานน้อยกว่าการทำงานแบบตามลำดับขั้นปกติ

ในงานวิจัย ParalleJS : การทำงานของจาวาสคริปต์บนระบบที่มีหลากหลายรูปแบบของ Jin Wang et al[9]. ได้มีการนำเสนอเฟรมเวิร์คในการนำไปทำงานบนระบบที่มีหลากหลายแบบคือสามารถทำงานบนซีพียู จีพียู โดยการพัฒนาโปรแกรมด้วยภาษาระดับสูงซึ่งผลลัพธ์ที่ได้มีความเร็วเพิ่มขึ้น 26.8 เท่าเมื่อเทียบกับการพัฒนาด้วยเว็บบราวเซอร์ที่ใช้จาวาสคริปต์ปกติ

ในงานวิจัยการประมวลผลแบบกระจาย (Distributed Computing) บนบราวเซอร์ของ Cushing, Reginald, et al[12]. ได้กล่าวถึงประสิทธิภาพของภาษาจาวาสคริปต์ การประมวลผลของเว็บบราวเซอร์ โดยใช้ WeevilScount และได้นำเอาเว็บบราวเซอร์มาทดสอบประสิทธิภาพของการประมวลผลในงานทางด้านชีวสารสนเทศ (bio-informatics) โดยใช้ภาษาจาวาสคริปต์ในการพัฒนาซึ่งในการประมวลผลงานได้อย่างมีประสิทธิภาพมากยิ่งขึ้น