บทนำ

เนื่องจากงานวิจัยด้านเคมีคอมพิวเตอร์ (computational chemistry) มีความหลากหลายและขยายกว้างมากขึ้น เช่น งานวิจัยทางด้านการออกแบบยา (drug design) การพัฒนาด้านซีโอไลต์ (zeolite) และอื่น ๆ เพราะแป็นวิธีที่มีประสิทธิภาพและช่วยลดค่าใช้จ่ายจากการทำการทดลอง อย่างไรก็ตามการใช้โปรแกรมทางการค้า (commercial programs) เช่น โปรแกรม Gaussian และโปแกรม Amber รวมถึงการใช้ cluster computer มีราคาแพงและอาจมีความยุ่งยากต่อการใช้งาน นอกจากนี้การเปลี่ยนแปลงที่รวดเร็วของฮาร์ดแวร์ (hardware) และซอร์ฟแวร์ (software) ทำให้ยากต่อการลงทุน เนื่องจากคอมพิวเตอร์จำนวนมากที่ใช้เป็น cluster computer จะล้าหลังภายในไม่กี่ปี

การสร้างระบบที่ประหยัด และมีประสิทธิภาพต่อการทำคอมพิวเตอร์โมเดลลิ่ง (computational modelling) อีกทั้งมีความยืดหยุ่นและคุ้มค่าต่อการเปลี่ยนแปลงที่รวดเร็วของฮาร์ดแวร์และเทคนิคการโปรแกรมใหม่ ๆ จึงมีความจำเป็น โดยงานวิจัยนี้จะเกี่ยวข้องกับการใช้ open-source software ที่มี source code สำหรับการทำคอมพิวเตอร์โมเดลลิ่ง เพื่อเป็นการประหยัดค่าใช้จ่ายในการซื้อซอร์ฟแวร์ โดยโปรแกรมเหล่านี้จะถูกนำไปใช้ในการโมเดลหรือคำนวณบน high-end 3D-graphics cards ที่มีประสิทธิภาพของคอมพิวเตอร์ส่วนบุคคล (PC) ซึ่งไม่กี่ปีที่ผ่านมาวีดีโอการ์ด (video cards) ได้ถูกพัฒนาให้มีประสิทธิภาพต่อระบบการดำเนินการแบบ parallel และสามารถประหยัดเวลาในการทำคอมพิวเตอร์โมเดลลิ่ง ได้ถึง 40% โดยคาดว่ามีความเร็วประมาณสิบเท่าของ Intel CPU ที่สามารถทำได้

โดยวัตถุประสงค์ของงานวิจัยนี้คือ การหา open-source softwares ที่เหมาะสมต่อวิธี MD-ONIOM ซึ่งเป็นวิธีที่ถูกใช้ในการสร้างโมเดลของโมเลกุลในตัวทำละลาย และพัฒนาซอร์ฟแวร์โดยการทำ re-programming เพื่อให้โปรแกรมมีความเหมาะสมและจำเพาะต่อการคำนวณบน high-end 3D-graphics cards โดยการใช้ GPGPUซึ่งจะทำให้ลดค่าใช้จ่ายของการทำคอมพิวเตอร์โมเดลลิ่ง เพราะเสมือนเป็นการสร้าง cluster-like บนเครื่องคอมพิวเตอร์ส่วนบุคคลเครื่องเดียว นอกจากนี้ยังทำให้เกิดความยืดหยุ่นต่อการเปลี่ยนแปลงอย่างรวดเร็วของฮาร์ดแวร์ และซอร์ฟแวร์ในท้องตลาดด้วย

โดยขอบเขตของงานวิจัยคือ การสร้างคอมพิวเตอร์ที่มีประสิทธิภาพด้วยการใช้ open-source softwares โดยการศึกษา source codes สำหรับ modelling software และ GPU stream programming จากนั้นนำ open-source softwares ไปใช้บน GPU และนำไปใช้กับวิธี MD-ONIOM ต่อไป

วัตถุประสงค์ของโครงการวิจัย

1. เพื่อหา open-source softwares ที่เหมาะสมต่อวิธี MD-ONIOM ซึ่งเป็นวิธีที่ถูกใช้ในการสร้างโมเดลของโมเลกุลในตัวทำละลาย

2. เพื่อพัฒนาซอร์ฟแวร์โดยการทำ re-programming เพื่อให้โปรแกรมมีความเหมาะสมและจำเพาะต่อการคำนวณบน high-end 3D-graphics cards โดยการใช้วิธี GPGPU
3. เพื่อลดค่าใช้จ่ายของการทำคอมพิวเตอร์โมเดลลิ่ง
4. เพื่อสร้าง cluster-like บนเครื่องคอมพิวเตอร์ส่วนบุคคล เครื่องเดียว

ประโยชน์ที่คาดว่าจะได้รับ

1. พัฒนาการใช้คอมพิวเตอร์ส่วนบุคคล เพียงเครื่องเดียวในการทำระบบโมเดลลิ่ง(modelling system) ที่มีประสิทธิภาพสูง และสามารถประยุกต์ใช้กับงานวิจัยอื่น ๆ
2. พัฒนาทักษะการโปรแกรม รวมถึงสร้างประสบการณ์กับวิธีการล่าสุดของการโปแกรม GPC's (stream programming)
3. เพิ่มความเข้าใจความสัมพันธ์ระหว่างระบบคอมพิวเตอร์และโมเดลลิ่งทางเคมี (chemistry modelling) และความเหมาะสมระหว่างค่าใช้จ่ายและเวลาที่ใช้ ในการสร้างระบบโมเดลลิ่งที่มีความถูกต้อง
4. พัฒนางานวิจัยโดยการใช้เทคนิคการโปแกรมล่าสุด เพื่อการดำเนินการข้อมูลแบบ parallel และเพิ่มประสบการณ์ในแง่ของการพัฒนางานวิจัยเพื่อไม่ให้ล้าหลัง
5. ความร่วมมือ และปรึกษาระหว่างหลายสาขา เช่น คณิตศาสตร์ วิทยาศาสตร์คอมพิวเตอร์ เคมี และวิทยาศาสตร์สาขาอื่น ๆ และหวังว่าจะนำไปสู่การเปิดสอนวิชาทางด้าน advanced computer programming สำหรับนิสิตสาขาวิทยาศาสตร์ต่อไป

วิธีวิจัย

ระบบที่ใช้ในการศึกษาทางเคมีสมัยใหม่ ไม่ว่าจะเป็นการศึกษาทางด้านโมเดล หรือการทดลอง มักมีความซับซ้อน และอาจเกี่ยวข้องกับโมเลกุลมากมาย รวมถึงขึ้นตอนการทำหลายขั้นตอน โดยเฉพาะการทำการทดลองที่อาจมีค่าใช้จ่ายสูงมาก แต่เนื่องจากความก้าวหน้าทางคอมพิวเตอร์ โดยเฉพาะคอมพิวเตอร์ส่วนบุคคล ทำให้เกิดการพัฒนาด้านการทำโมเดลลิ่ง และซิมูเลชั่น (simulations) เป็นอย่างมาก [1-2] การเกื้อกูลกันระหว่างการ ทำโมเดลลิ่ง และการทำการทดลอง ทำให้เกิดความเข้าใจระบบที่ศึกษามากขึ้น ประหยัด และยังช่วยชี้นำให้เห็นว่าการทดลองแบบใดที่จะนำไปสู่ตามสมมุติฐานที่ตั้งไว้ ตัวอย่างที่สำคัญของงานวิจัยในประเทศไทย คือ การออกแบบยา ที่ใช้ โมเดลลิ่งในการศึกษาหาประสิทธิภาพการต้าน (inhibition) ของโมเลกุลกับเอ็นไซน์ที่สนใจ โดยสามารถทดสอบกับสารประกอบจำนวนมากที่มีอยู่ในฐานข้อมูล ด้วยเวลาที่เร็วกว่าและประหยัดกว่าการทดลองในห้องปฏิบัติการเป็นอย่างมาก [3-4]

Molecular Dynamics Simulations (MD) [5] สามารถใช้ในการทำซิมูเลชั่น (simulate) โมเลกุลแบบไดนามิกส์ (dynamics) ทั้งในตัวทำละลาย หรือในสภาวะก๊าซ และเหมาะอย่างยิ่งสำหรับระบบที่มีขนาดใหญ่ เช่น โปรตีน ซึ่งสามารถแสดงให้เห็น dynamics trajectories หรืออิทธิพลของตัวทำละลายได้ หากระบบที่ศึกษามีขนาดไม่ใหญ่มากนัก และต้องการความถูกต้องที่มากขึ้นก็สามารถโมเดลโดยใช้ทฤษฎีควอนตัม (Quantum Mechanics, QM) [6-7] ในการคำนวณโมเดล จากงานที่ผ่านมาของผู้ขอทุนวิจัยได้เสนอระเบียบวิธี MD-ONIOM2 (molecular dynamics simulations and Our own N-layered Integrated molecular Orbital and Molecular mechanics approach) [8-9] โดยเป็นการโมเดลสองขึ้นตอน คือ MD และ QM เพื่อใช้สำหรับการคำนวณโปรตอนเอ็นเอ็มอาร์ (^{1}H-NMR) ของโปรตอนที่มีความว่องไว (acidic proton) ซึ่งประสบความสำเร็จเพราะสามารถคำนวณ ^{1}H-NMR ได้อย่างถูกต้อง และกำลังถูกพัฒนาต่อไป

ปัญหาหลักของการทำซิมูเลชั่น คือ ข้อจำกัดคอมพิวเตอร์ [10] การที่จะได้ผลที่มีความถูกต้องต้องการโมเดลที่เหมือนจริง นั่นคืออาจมีจำนวนของอะตอมมากมาย และจะคำนวณรายละเอียดของอันตรกิริยาต่าง ๆ ที่มีในระบบ ดังนั้นการทำคำนวณโดยใช้โมเดลลิ่ง จึงต้องพิจารณาทั้งในแง่ของความถูกต้องของโมเดลที่ต้องการกับขีดจำกัดที่เราจะสามารถทำได้ อย่างไรก็ตาม Moore's Law [11] ชี้ให้เห็นว่าจำนวนของทรานซิสเตอร์ (transistor) ที่มีอยู่ใน CPU ของคอมพิวเตอร์เพิ่มขึ้นเป็น 2 เท่าทุก ๆ 2 ปี ดังนั้นประสิทธิภาพของคอมพิวเตอร์จึงเพิ่มมากขึ้นเป็นแบบ exponential rate และนั่นก็ทำให้งานทางด้านคอมพิวเตอร์โมเดิลลิ่ง สามารถทำได้กับระบบที่ซับซ้อน และมีความเหมือนจริงมากขึ้น

เมื่อ 2-3 ปีที่ผ่านมา 3D-graphics cards สำหรับคอมพิวเตอร์ส่วนบุคคลได้ถูกพัฒนาขึ้นอย่างรวดเร็ว โดยเฉพาะอย่างยิ่งสำหรับเกมส์บนคอมพิวเตอร์ในการแสดงภาพ Graphics Processing Unit (GPU) สำหรับ 3D-graphics cards จึงถูกพัฒนาขึ้นเร็วกว่า Central Porcessing Unit (CPU) [12-13] ปัจจุบัน G80 GPU จาก

NVIDIA (ตัวอย่างเช่น GeForce 8800 GTX graphics card, US$570) สามารถปฏิบัติการ 350GFLOPS เมื่อเปรียบเทียบกับ 40GFLOPS Intel 3.0GHz Core2Duo CPU (US$266) ดังนั้น graphics card จึงมีความสามารถในการคำนวณได้ประมาณ 10 เท่าของ CPU ของคอมพิวเตอร์ที่มีประสิทธิภาพสูง $0.95/GFLOPS ต่อ $6.65/GFLOPS และมีประสิทธิสูงเมื่อใช้คำนวณทาง computer modelling บนคอมพิวเตอร์PC

ในระยะเวลา 10 ปีที่ผ่านมา GPU ได้ถูกพัฒนาจากเครื่องมือที่ใช้ประโยชน์เฉพาะทางสำหรับภาพบนจอคอมพิวเตอร์ ไปสู่เครื่องมือที่สามารถถูกโปรแกรมได้สำหรับใช้งานด้านอื่น ๆ ที่บรรจุไปด้วยมากกว่า 128 processing units ในการปฏิบัติการแบบ parallel data processing อย่างมีประสิทธิภาพ [10] และใน 2-3 ปีที่ผ่านมา GPU ถูกใช้กว้างขวางขึ้น รวมทั้งในด้านการคำนวณทางวิทยาศาสตร์ ปัจจุบัน โปรแกรมชั้นสูงต่าง ๆ ประเภท C-like languages (ตัวอย่างเช่น Cg, BrookGPU, Sh) [14-16] ก็เอื้ออำนวยต่อการนำ algorithms ไปใช้แบบ parallel processing และใช้ประโยชน์ได้หลากหลายรวมถึงการทำ MD simulations ที่แสดงให้เห็นว่าสามารถซิมูเลตได้เร็วกว่า 10 ถึง 100 เท่า จากการใช้ GPU [10, 17]

ในโครงการวิจัยนี้จะเป็นการพัฒนาแนวคิดและประสบการณ์ ในงานด้านดังกล่าวและจะนำไปประยุกต์ใช้กับวิธี MD-ONIOM โดยการดำเนินการบน GPU เพื่อที่จะพัฒนาการคำนวณให้เร็วขึ้น จะเป็นประโยชน์ให้กับกลุ่มนักวิจัยในประเทศไทยได้ใช้เทคโนโลยีที่ราคาประหยัดขึ้น สำหรับการฝึก การสอน และการทำวิจัย

วิธีการดำเนินการวิจัย

1. หาฮาร์ดแวร์ที่เหมาะสมสำหรับเครื่องคอมพิวเตอร์แบบ PC รวมถึง CPU, RAM และกราฟฟิคการ์ด (graphics card)
2. หาซอร์ฟแวร์ที่เหมาะสมสำหรับ N Vidia และ ATI graphics cards
3. เลือก open-source software ที่เหมาะสมสำหรับ MD-ONIOM modelling
4. สร้างระบบและทดสอบโดยใช้โมเดลอย่างง่ายเพื่อยืนยันว่า ระบบที่สร้างใช้ได้
5. เลือก software code ที่สนใจที่จะเปลี่ยนและพัฒนาในการดำเนินการบน GPC โดยเฉพาะอย่างยิ่งส่วนที่ช้าที่สุดในการคำนวณ
6. ร่วมปรึกษากับผู้เชี่ยวชาญในสาขาต่าง ๆ เช่น คณิตศาสตร์ และวิทยาศาสตร์คอมพิวเตอร์ เพื่อหาแนวทางที่มีประสิทธิภาพที่สุดต่อการดำเนินการด้วย กราฟฟิคการ์ด

ผลการวิจัย

ผลงานวิจัยตามแผนการดำเนินงานที่ได้นำเสนอในแบบเสนอโครงการวิจัย (research project) รายงานผลการวิจัยเป็นส่วน ๆ ดังนี้

ส่วนประกอบฮาร์ดแวร์

ระบบของคอมพิวเตอร์ที่จะสร้างขึ้นสำหรับโครงการนี้ จะต้องประกอบไปด้วยฮาร์ดแวร์ที่มีประสิทธิภาพสูงที่สามารถใช้ในการคำนวนได้อย่างมีประสิทธิภาพ อย่างไรก็ตามประสิทธิภาพนั้นจะต้องมีความเหมาะสมกับราคาของฮาร์ดแวร์ด้วย ดังนั้นฮาร์ดแวร์ที่เลือกใช้จึงไม่ใช่ฮาร์ดแวร์ที่มีประสิทธิภาพสูงที่สุด แต่เป็นฮาร์ดแวร์ที่มีประสิทธิภาพสูงในระดับที่เพียงพอที่จะทำงานวิจัยได้ดี และมีแตกต่างของราคาและประสิทธิภาพที่ยอมรับได้เมื่อเทียบกับฮาร์ดแวร์ที่มีประสิทธิภาพสูงสุด ณ เวลาที่ทำการวิจัย โดยส่วนประกอบฮาร์ดแวร์ที่สำคัญที่สุดมีดังนี้

1. CPU โดย CPU ที่ถูกเลือกสำหรับงานวิจัยนี้คือ Intel Q9450 Core 2 Quad ซึ่งประกอบไปด้วย 4 cores ที่สามารถทำงาน หรือคำนวณโดยทั่วไปแบบ parallelism บน CPU และมีประสิทธิภาพสูงเกือบเทียบเท่ากับความสามารถที่เราคาดหวังว่าคอมพิวเตอร์ส่วนบุคคลจะทำได้

2. Nvidia GTX260 graphics card ได้ถูกเลือกมาใช้ในงานวิจัยนี้ เนื่องจากเป็น GPU รุ่นล่าสุดของ “Nvidia” ณ ขณะที่ทำงานวิจัย แต่อย่างไรก็ตามองค์ประกอบของระบบคอมพิวเตอร์อื่น ๆ กำลังอยู่ในช่วงที่กำลังพิจารณาว่าจะต้องเป็นอย่างไรเพื่อมารองรับ GPU รุ่นนี้ ทางโครงการวิจัยฯ จำเป็นต้องรอจนสามารถสั่งซื้อ Nvidia GTX260 graphics card ได้ในประเทศไทย เนื่องจากการนำเข้าของ Nvidia GTX260 graphics card ค่อนข้างจะช้ากว่าในต่างประเทศ จากนั้นจึงได้ออกแบบองค์ประกอบของระบบคอมพิวเตอร์อื่น ๆ เพื่อมาใช้งานต่อไป

3. Motherboard ที่เลือกใช้ในงานวิจัยเป็นรุ่นเฉพาะล่าสุด คือ PCIe2.0 เป็น High-performance bus system สำหรับบัตร (cards) ที่สามาถใส่เข้าไปในระบบคอมพิวเตอร์ โดยเฉพาะอย่างยิ่ง motherboard รุ่นนี้สามารถถ่ายเทข้อมูลระหว่างกราฟิค การ์ด กับ CPU/RAM ได้รวดเร็วมาก เนื่องจากการถ่ายเทข้อมูลจะเป็นส่วนที่ช้าที่สุดสำหรับการคำนวณโดย GPU จึงทำให้การเลือก motherboard ที่เหมาะสมและมีประสิทธิภาพมีความสำคัญอย่างมากในงานวิจัยนี้

ในส่วนขององค์ประกอบอื่น ๆ ของระบบคอมพิวเตอร์ที่เลือกใช้ เช่น RAM hard-drive และ DVD-ROM ได้เลือกใช้แบบมาตรฐานทั่วไป เนื่องจากองค์ประกอบเหล่านี้ไม่มีผลกระทบอย่างมีนัยสำคัญต่อการคำนวณ

ส่วนประกอบซอร์ฟแวร์

ระบบของคอมพิวเตอร์ที่ทางโครงการวิจัยฯ ได้สร้างขึ้นสามารถดำเนินการได้ทั้งระบบ ไมโครซอร์ฟ วินโดวส์ (MS Windows) และระบบปฏิบัติการแบบลีนุกซ์ (Linux operation system) ทางโครงการวิจัยฯได้ดาวน์โหลด CUDA2 จาก Nvidia website และทำการติดตั้ง CUDA2 บนคอมพิวเตอร์ เนื่องจาก CUDA2 มีเครื่องมือ (tools) ที่จำเป็นในการควบคุมการเขียนโปรแกรมบนกราฟฟิค การ์ด ซึ่งเป็นการต่อยอดของโปรแกรมมาตรฐานภาษา C++ นั่นคือเราสามารถเขียนโปรแกรมภาษา C++ ด้วยคำสั่ง และโครงสร้างทางข้อมูลได้พิเศษขึ้นในการทำโปรแกรม และยังสามารถถูก complied ได้ในแบบปกติ

โปรแกรมที่เขียนด้วยโปรแกรมภาษา C++ ได้ถูก complied และทดสอบโดยการดำเนินการบนระบบคอมพิวเตอร์ที่โครงการวิจัยฯ ได้สร้างขึ้น เพื่อทำการทดสอบกราฟฟิค การ์ด ว่าสามารถทำงานได้ตามคำสั่ง ซึ่งพบว่าการทดลองนี้เป็นผลสำเร็จ ให้ผลการทดลองที่ถูกต้องเรียบร้อยแล้วบนคอมพิวเตอร์ แสดงให้เห็นว่าระบบคอมพิวเตอร์ที่ได้สร้างขึ้นนี้สามารถดำเนินการได้ตามที่คาดหวังไว้

การโปรแกรม

เนื่องจาก General-Purpose computation on Graphics Processing Units (GPGPU) มีการพัฒนาขึ้นอย่างรวดเร็ว โดยในเดือนสิงหาคม พ.ศ. 2551 Intel GPU system ชื่อ “Larrabee” ได้ถูกนำเสนอ และได้ถูกยกเลิกไปในเดือนพฤษภาคม พ.ศ. 2553 จึงทำให้เฉพาะ Nvidia และ AMD/ATI graphics cards เท่านั้นที่สามารถใช้ได้กับ GPGPU

ในเดือน สิงหาคม พ.ศ. 2552 OpenCL (programming language) ได้ถูกนำเสนอขึ้นมาเพื่อสามารถโปรแกรม Nvidia และ AMD/ATI graphics cards และได้มีการพัฒนาขึ้นอีกในนเดือนมิถุนายน พ.ศ. 2553 ได้มีการถกเถียงในสังคม GPGPU ออนไลน์เรื่อยมา ในเรื่องของการเรียนรู้และการใช้ OpenCL สำหรับ GPU หรือ CUDA สำหรับ Nvidia เท่านั้น ว่าสิ่งไหนดีกว่ากัน

การเลือกใช้ open-source modelling software

ทางโครงการวิจัยฯสามารถถูกประยุกต์ใช้ GPU programming สำหรับการทำโปรแกรมโมเดลลิ่งทางเคมีเรียบร้อยแล้ว และได้พิจารณาถึงโปรแกรมทางเคมีที่จะนำมาประยุกต์ใช้ และผู้วิจัยก็ได้เข้าร่วมการสัมมนาเรื่อง General-Purpose computation on Graphics Processing Units (GPGPU) ณ ห้องประชุม 110 (Theatre) อาคารส่วนงานกลาง สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ เพื่อแลกเปลี่ยนความรู้ ความคิดเห็นกับนักวิจัยท่านอื่น ๆ ที่ทำงานอยู่ในด้านนี้ และได้ข้อมูลว่าจะมีโปรเจคชื่อ “OpenMM” (https://simtk.org/home/openmm/) [18] ออกมา ซึ่ง “OpenMM” นี้จะมีเครื่องมือมากมาย (a free library of

tools) ที่จะสามารถใช้กับโปรแกรมทางโมเดลลิ่งอื่น ๆ ได้ เช่น GROMACS ที่เป็น open-source modelling software ที่จะใช้ GPU ในการเร่งการคำนวณให้มีความรวดเร็วขึ้น

เมื่อไม่นานมานี้ได้มีการทดสอบโปรแกรมทางโมเดลลิ่ง เพื่อที่จะค้นหาว่ามี open source package ใดที่เหมาะสม เนื่องจาก open source นี้มีความสำคัญมากเนื่องจากเราสามารถที่จะปรับปรุงโค้ดของโปรแกรม (program code) ดั้งเดิมในการทำการทดลอง และสามารถนำมาใช้ได้โดยไม่ต้องซื้อลิขสิทธิทธิ์หรือมีค่าใช้จ่ายใด ๆ โปรแกรม GROMACS และ NAMD เป็น open-source modelling software ที่น่าสนใจที่ทางโครงการวิจัยฯ จะดำเนินงานวิจัยต่อไป

“OpenMM” ที่สามารถใช้ในโปรแกรม GROMACS ร่วมด้วย GPU ที่สนับสนุนได้ถูกนำเสนอขึ้นในเดือนมกราคม พ.ศ. 2552 จากนั้น Beta 1.0 ได้ถูกนำเสนอขึ้นในเดือนตุลาคม พ.ศ. 2553 เพื่อเป็นการทดลอง และยังคงมีการปรับปรุงการใช้งานอย่างตลอดเวลา

สำหรับการพัฒนาของโปรแกรม NAMD พบว่าในเวอร์ชั่น 2.6 ที่ได้ถูกนำเสนอในเดือนมีนาคม พ.ศ. 2551 ยังคงไม่มี GPU แต่ต่อมาในเวอร์ชั่น 2.7 ที่ถูกนำเสนอออกมาในเดือนมกราคม พ.ศ. 2553 ได้มีการพัฒนาโดยมี CUDA/Nvidia GPU ในโปรแกรมด้วย และยังคงมีการปรับปรุงการใช้งานอย่างตลอดเวลา

นอกจากนี้กลุ่มนักวิจัยดั้งเดิมที่พัฒนาด้านโปรแกรม และโมเดลลิ่งทางเคมีได้เริ่มที่จะใช้ GPGPU และทำให้สามารถทำให้เราใช้งานได้ในโปรแกรม GROMACS และ NAMD ดังนั้นการโปรแกรม GPU จึงเริ่มเป็นมาตรฐานสำหรับการทำโมเดลลิ่งทางเคมี โดยเฉพาะ graphics cards แต่ละชนิดต้องการการแก้ไขพารามิเตอร์ (tuning) เนื่องจากแต่ละชนิดมีขนาดความจำ และจำนวน cores ต่างกัน

สรุป วิจารณ์ และข้อเสนอแนะ

โครงการวิจัย การใช้ Open-Sources Software สำหรับวิธี MD-ONIOM โดยการใช้ GPGPU เพื่อเพิ่มประสิทธิภาพการโมเดลให้ดียิ่งขึ้น ได้ดำเนินงานเป็นไปตามแผน โดยได้มีการเลือกฮาร์ดแวร์ที่ใช้เป็นองค์ประกอบสำคัญในการประกอบเป็นคอมพิวเตอร์ที่มีประสิทธิภาพสูง และเหมาะสมกับการทำการวิจัย รวมถึงได้ซอร์ฟแวร์ที่มีประสิทธิภาพในการใช้งานเรียบร้อยแล้ว โดยคำนึงถึงการประหยัดและความคุ้มค่า นอกจากนี้ยังได้ทำการทดสอบระบบเพื่อยืนยันว่าระบบคอมพิวเตอร์ที่ได้สร้างมาสามารถทำงานตามคำสั่งเป็นผลสำเร็จได้จริงตามความคาดหมาย เพื่อที่ที่จะได้ดำเนินงานวิจัยกับ open-source modelling software ที่เหมาะสมในลำดับต่อไป

โดยผู้วิจัยในโครงการวิจัยฯ พยายามที่จะร่วมปรึกษากับผู้เชี่ยวชาญทั้งในทางสาขาคอมพิวเตอร์และสาขาอื่น ๆ ที่เกี่ยวข้อง อย่างที่ผ่านมาผู้ทำวิจัยในโครงการวิจัยฯ นี้ได้เข้าร่วมสัมมนาเรื่อง “General-Purpose computation on Graphics Processing Units (GPGPU)” เพื่อที่จะพัฒนางานวิจัยชิ้นนี้ให้มีประโยชน์สูงสุด และประยุกต์ใช้ได้หลากหลายอย่างมีประสิทธิภาพ อย่างไรก็ตามพบว่านักวิจัยในประเทศไทยส่วนน้อยที่ทำงานวิจัยทางด้านนี้ ทั้ง ๆ ที่ในต่างประเทศให้ความสำคัญ และมีผลงานวิจัยออกมาอย่างต่อเนื่อง เนื่องจากผลงานที่ได้สามารถช่วยลดค่าใช้จ่ายในการคำนวณ และยังเป็นผลงานตีพิมพ์ได้อย่างดี

อย่างไรก็ตามทางโครงการวิจัยฯ จะหาแนวทางที่ทำให้ GPGPU สามารถใช้สำหรับการทำการคำนวณโดยไม่ซ้ำกับงานในกลุ่มงานวิจัยอื่น ๆ ซึ่งจะเป็นแนวที่เกี่ยวข้องกับการทำให้ง่ายขึ้นต่อการโปรแกรม ทำให้งานวิจัยต่าง ๆ สามารถเริ่มคำนวณได้รวดเร็วขึ้นโดยไม่ต้องเสียเวลามากกับการเตรียมข้อมูลสำหรับการคำนวณ หรือแนวทางที่เกี่ยวข้องกับการประยุกต์การคำนวณทางทฤษฎีควอนตัมบน GPU และการคำนวณของ Molecular Dynamics Simulation (MD) และ Molecular Mechanic Simulation (MM)

ทั้งนี้เนื่องจากกลุ่มนักวิจัยดั้งเดิมที่พัฒนาด้านโปรแกรม และโมเดลลิ่งทางเคมีได้เริ่มที่จะใช้ GPGPU และทำให้สามารถทำให้เราใช้งานได้ในโปรแกรม GROMACS และ NAMD ดังนั้นการโปรแกรม GPU จึงเริ่มเป็นมาตรฐานสำหรับการทำโมเดลลิ่งทางเคมี โดยเฉพาะ graphics cards แต่ละชนิดต้องการการแก้ไขพารามิเตอร์ (tuning) เนื่องจากแต่ละชนิดมีขนาดความจำ และจำนวน cores ต่างกัน

เอกสารอ้างอิง

1. J. A. McCammon, B. R. Gelin, M. Karplus, Nature 267 (1997), 585.
2. K. Y. Sanbonmatsu, S. Joseph, C. Tung, PNAS 102 (2005) 15854.
3. S. Saen-oon, S. Hannongbua, P. Wolschann, J. Chem. Inf. Comput. Sci. 43 (2003) 1412.
4. S. Hannongbua, S. Prasithichokekul, P. Pungpo, Compt. Aided Mol. Des. 15 (2001) 997.
5. M. Svensson, S. Humbel, R.D.J. Froese, T. Matsubara, S. Sieber, K. Morokuma, J. Phys. Chem. 100 (1996) 19357.
6. P.B. Karadakov, L. Morokuma, Chem. Phys. Lett. 317 (2000) 589.
7. S. Dapprich, I. Komaromi, K.S. Byun, K. Morokuma, M.J. Frisch, J. Mol. Struct. (Theochem) 461 (1999) 1.
8. V. Vailikhit, P. Bunsawansong, S. Techasakul, S. Hangnongbua, J. Theor. Comp. Chem. 5 (2006) 913.
9. V. Vailikhit, W. Treesuwan, S. Hannongbua, J. Mol. Struct. (THEOCHEM) 806 (2007) 99.
10. J. E. Stone, J. C. Phillips, P. L. Freddolino, D. J. Hardy, L. G. Trabuco, K. Schulten, J. Comp Chem 28 (2007) 2618.
11. G. E. Moore, Electronics, 38 (1965) 114.
12. J. D. Owens, D. Luebke, N. Govindaraju, M. Harris, J. Krüger, A. E. Lefohn, T. J. Purcell, Computer Graphics Forum, 26 (2007) 80.
13. L. Marziale, G. G. Richard, V. Roussev, Digital Investigation, 4 (2007) 73.
14. http://graphics.stanford.edu/projects/brookgpu/
15. http://libsh.org/index.html
16. http://arxiv.org/PS_cache/arxiv/pdf/0706/0706.3060v1.pdf
17. M. S. Friedrichs, P. Eastman, V. Vaidyanathan, M. Houston, S. LeGrand, A. L. Beberg, D. L. Ensign, C. M. Bruns, V. S. Pande, J. Comp. Chem, 30(6) (2009) 864.