

การประมวลผลบนคลาวด์ : โครงสร้างพื้นฐานสำหรับ Internet of Things

Cloud Computing: Infrastructure for the Internet of Things

ชลธิป ทุมกานนท์¹, สุธรรม อัสวาศักดิ์สกุล¹, นิกอร์ โภคอุดม¹ และกัญญ์ณิชา โภคอุดม²
Chalathip Thumkanon¹, Sutharm Asavasaksakul¹, Nikorn Pokudom¹ and Kannicha Pokudom²

¹คณะวิศวกรรมศาสตร์ มหาวิทยาลัยอีสเทิร์นเอเซีย

¹Faculty of Engineering, Eastern Asia University

²คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยอีสเทิร์นเอเซีย

²Faculty of Information Technology, Eastern Asia University

บทคัดย่อ

Internet of Thing - IoT เป็นแนวคิดใหม่ที่สำคัญในการพัฒนาเทคโนโลยีอัจฉริยะ ที่มีจุดหมายในการสร้างอุปกรณ์ชาญฉลาด ช่วยให้การดำเนินชีวิตของเราสะดวกมากขึ้น การประมวลผลบนคลาวด์สามารถปลดปล่อยศักยภาพที่แท้จริงของการใช้งานอินเทอร์เน็ตของ IoT บทความนี้จะกล่าวถึงการวิวัฒนาการของระบบประมวลผลแบบคลาวด์และแบบจำลองในการให้บริการของคลาวด์ และชี้ให้เห็นถึงศักยภาพที่แท้จริงของ IoT เมื่อใช้งานระบบประมวลผลบนคลาวด์

คำสำคัญ: ไอโอที, อินเทอร์เน็ตออฟริงส์, การประมวลผลคลาวด์, การประมวลผลบนคลัสเตอร์

Abstract

Internet of Things (IoT) is a recent concept that has a big impact on smart technology which aims to invent smart devices that support our way of life. Cloud computing can unleash the real potential of Internet to IoT. This article explains the evolution of computer and Cloud computing. It points out the real potential that IoT devices could have when using cloud computing.

Keywords: IoT, internet of things, cloud computing, cluster computing



บทนำ

Internet of Things -IoT เป็นแนวคิดที่ว่าอุปกรณ์ชาญฉลาด (smart devices) จะมีความฉลาดหรือมีประโยชน์หลากหลายมากขึ้นถ้าอุปกรณ์เหล่านั้นเชื่อมต่อเข้าสู่อินเทอร์เน็ตได้ อุปกรณ์ชาญฉลาดที่กล่าวถึงก็คืออุปกรณ์ที่มีระบบคอมพิวเตอร์ฝังตัว (embedded system) ผนวกอยู่ด้วย (จริง ๆ แล้วคอมพิวเตอร์ก็นับได้ว่าเป็นอุปกรณ์ชาญฉลาดอย่างหนึ่ง แต่การเชื่อมต่อเข้าสู่อินเทอร์เน็ต

เป็นเรื่องปกติของคอมพิวเตอร์อยู่แล้ว) และแนวความคิดนี้ก็ได้ขยายไปสู่อุปกรณ์ธรรมดาหรือเครื่องใช้ไฟฟ้าทั่วไป รวมถึงระบบอัตโนมัติต่าง ๆ เช่น ระบบบ้านอัจฉริยะ ระบบรักษาความปลอดภัยและกล้องวงจรปิด เป็นต้น

IoT ทำให้อุปกรณ์ฉลาดขึ้นได้อย่างไร? เพื่อตอบคำถามนี้ ขอยกตัวอย่างอุปกรณ์ต่าง ๆ ที่เมื่อต่อเข้ากับอินเทอร์เน็ตแล้วจะมีประโยชน์ใช้งานมากขึ้น

ตัวอย่างแรก อุปกรณ์นำทาง (navigation system) ซึ่งเป็นอุปกรณ์ชาญฉลาดอยู่แล้ว ใช้ข้อมูลจากดาวเทียมเพื่อหาพิกัด สามารถคำนวณหาเส้นทางไปสู่จุดหมายที่สั้นที่สุด สามารถบอกทิศทางโดยใช้เสียง คำนวณระยะทางและเวลาในการเดินทาง และถ้าเชื่อมต่อเข้าสู่อินเทอร์เน็ตอาจจะแสดงความหนาแน่นของการจราจร และคำนวณหาเส้นทางใหม่เพื่อเลี่ยงการจราจรที่ติดขัดได้ตามเวลาจริง

ตัวอย่างที่สอง อุปกรณ์ธรรมดาเช่น ตู้เย็น หากมีระบบคอมพิวเตอร์ฝังตัว อาจช่วยบันทึกรายการอาหารที่อยู่ในตู้ และคอยเตือนเมื่อมีขึ้นไหนจะหมดอายุ และถ้าเชื่อมต่ออินเทอร์เน็ตได้ อาจช่วยให้เราจัดการสั่งซื้อหรือจองอาหารเพื่อแทนชิ้นที่หมดหรือหมดอายุจากร้านค้าได้โดยตรง

Cloud Computing หรือการคำนวณบนคลาวด์ (Tung-Hui, 2015; Qusay, 2011) เป็นการแบ่งปันทรัพยากรของคอมพิวเตอร์ที่เชื่อมต่อกันผ่านทางอินเทอร์เน็ตโดยทรัพยากรนั้นอาจจะเป็นพลังในการคำนวณ พื้นที่และความปลอดภัยในการจัดเก็บข้อมูล การรับส่งและประมวลผลข้อมูลตามเวลาจริง ฯลฯ

วิวัฒนาการของคอมพิวเตอร์และระบบประมวลผลบนคลาวด์ (Millard, 2013)

คอมพิวเตอร์ได้รับการพัฒนาขึ้นมาเพื่อตอบสนองความต้องการการประมวลผลเพื่อใช้แก้ไขปัญหาดังต่าง ๆ ทางด้านวิทยาศาสตร์และวิศวกรรม ที่สามารถจำลองปัญหาให้เป็นแบบจำลองทางคณิตศาสตร์ได้ (อยู่ในรูปแบบของโปรแกรมคอมพิวเตอร์) เช่น การพัฒนา ENIAC เพื่อคำนวณวิถีโค้งของการยิงขีปนาวุธ เป็นต้น การทำงานของคอมพิวเตอร์ในยุคแรกนี้จะประมวลผลทีละคำสั่งทีละข้อมูล (SISD--Single Instruction Single Data) ซึ่งแบบจำลองเหล่านี้วันก็จะซับซ้อนขึ้น จำเป็นต้องใช้พลังในการประมวลผลที่มากขึ้น และแม้ว่าปัจจุบันจะมีการใช้คอมพิวเตอร์ในด้านอื่น ๆ ที่ไม่ได้เกี่ยวข้องกับงานทางด้านวิทยาศาสตร์และวิศวกรรมโดยตรง แต่งานเหล่านั้นในที่สุดก็ต้องถูกจำลองมาเป็นโปรแกรมที่ต้องการพลังในการประมวลผลที่สูงเช่นกัน ดังนั้น การพัฒนาประสิทธิภาพในการประมวลผลจึงเป็นจุดมุ่งหมายหลักในการพัฒนา

คอมพิวเตอร์ตลอดมา

แนวคิดแรก ๆ ในการพัฒนาคอมพิวเตอร์ที่มีความสามารถในการประมวลผลสูง (high performance computing) ก็คือ การพัฒนาคอมพิวเตอร์ที่มีหน่วยประมวลผลเดี่ยว (single processor computer) โดยเน้นที่การพัฒนาและออกแบบหน่วยประมวลผลกลาง (CPU--Central Processing Unit) รวมทั้งระบบสนับสนุนต่าง ๆ เช่น หน่วยความจำ บัส ฯลฯ ที่มีความรวดเร็ว เช่น การสร้างซูเปอร์คอมพิวเตอร์ Cray และ Maspar เป็นต้น โดยหน่วยประมวลผลในยุคนี้จะมีเทคนิคในการเพิ่มความสามารถในการประมวลผล เช่น มีการทำงานแบบใช้คำสั่งเดียวกับข้อมูลจำนวนมาก (SIMD--Single Instruction Multiple Data) เพื่อใช้บวกหรือลบเวกเตอร์สองเวกเตอร์เข้าด้วยกันในคำสั่งเดียว (ไม่ต้องวนลูปเพื่อบวกหรือลบจำนวนเดียว (scalar) แต่ละจำนวน) เป็นต้น และแม้ว่าแนวคิดดังกล่าวจะได้รับการยอมรับว่าใช้ได้ แต่ก็มีค่าใช้จ่ายที่สูงมาก มีองค์กรเพียงไม่กี่แห่งเท่านั้นที่จะมีซูเปอร์คอมพิวเตอร์ได้ จึงมีแนวคิดในการสร้างระบบคอมพิวเตอร์ที่มีหลายหน่วยประมวลผล (multiple processor computer) โดยหน่วยประมวลผลแต่ละตัวมีประสิทธิภาพและราคาไม่สูงนัก แต่โดยรวมแล้วจะทำให้ได้คอมพิวเตอร์ที่มีประสิทธิภาพการประมวลผลที่สูงกว่าในราคาที่ถูกลงกว่าเดิม และก็เป็นจุดเริ่มต้นของการวิวัฒนาการการประมวลผลแบบขนาน (parallel computing evolution) มีการพัฒนาเทคนิคการเขียนโปรแกรมแบบหลายหน่วยประมวลผลหรือหลายเธรด (multiprocessor/ multithread programming) มีการสร้างระบบคอมพิวเตอร์แบบหลายหน่วยประมวลผลนี้ขายในเชิงพาณิชย์ ทำให้หลายองค์กรสามารถเป็นเจ้าของได้ แต่คอมพิวเตอร์เหล่านี้ก็ยังนับว่ามีราคาค่อนข้างสูงเมื่อเทียบกับเครื่องคอมพิวเตอร์ส่วนบุคคลทั่วไป

ต่อมา ก็มีแนวคิดในการใช้เครื่องคอมพิวเตอร์หลาย ๆ เครื่องต่อผ่านกันทางเครือข่าย เพื่อช่วยกันประมวลผล เครื่องเหล่านี้จะต้องเป็นเครื่องที่เหมือนกัน (homogenous) ที่ต่อผ่านอุปกรณ์เครือข่ายเฉพาะที่มีความเร็วสูง (high speed network) และเริ่มมีการใช้คำศัพท์เพื่อเรียกกลุ่มของเครื่องคอมพิวเตอร์ที่ช่วยกันเหล่านี้ว่า ฟาร์มหรือคลัสเตอร์ (computer farm / computer cluster) (Michael Armbrust, et al., 2010) และเครื่องแต่ละเครื่อง

ในกลุ่มจะเรียกว่าโหนด (node) คลัสเตอร์เหล่านี้มีประสิทธิภาพการประมวลผลที่สูง แต่ต้องมีการจัดการดูแลและต้องการอุปกรณ์ที่พิเศษ ต้องใช้เครื่องที่เหมือนกันเป็นจำนวนมากที่ต่ออยู่กับอุปกรณ์เครือข่ายพิเศษความเร็วสูง ทำให้มีเพียงบางสถาบันเท่านั้นที่สามารถพัฒนาและใช้ระบบแบบนี้ได้ ต่อมาได้มีการพัฒนาสร้างคลัสเตอร์ที่ผสมผสาน (heterogeneous) โดยเครื่องในคลัสเตอร์อาจจะเหมือนกันหรือแตกต่างกันก็ได้ ประกอบกับการที่อุปกรณ์เครือข่ายทั่วไปที่มีความเร็วสูงขึ้น จึงเป็นยุคเฟื่องฟูของการสร้างและใช้งานคอมพิวเตอร์ที่มีความสามารถในการประมวลผลสูงด้วยคลัสเตอร์ จากแต่เดิมที่เพียงบางสถาบันที่มีคลัสเตอร์เป็นของตน มาเป็นแทบทุกสถาบัน (โดยเฉพาะสถาบันการศึกษาระดับมหาวิทยาลัย) ต่างก็มีคลัสเตอร์ของตนและจนกลายเป็นหลายสถาบันมีคลัสเตอร์ร่วมกัน หรือหนึ่งคลัสเตอร์หลายสถาบัน (โหนดกระจายอยู่ในหลายสถาบัน) การประมวลผลของคลัสเตอร์นี้จะเป็นการประมวลผลแบบหลายคำสั่งบนหลายข้อมูลแตกต่างกัน (MIMD--Multiple Instruction Multiple Data) และแต่ละโหนดอาจใช้หน่วยประมวลผลที่ต่างรุ่นต่างยี่ห้อ หรือแม้แต่ต่างสถาปัตยกรรมทำงานร่วมกัน อาจทำให้ประสิทธิภาพไม่เท่ากัน (รวมทั้งความเร็วที่แตกต่างกันของเครือข่ายที่แต่ละโหนดเชื่อมต่ออยู่) จึงมีความต้องการที่จะบริหารจัดการให้คลัสเตอร์มีประสิทธิภาพที่เหมาะสม ทำให้เกิดการพัฒนาซอฟต์แวร์และเทคนิคที่เกี่ยวข้องมากมาย เช่น เทคนิคการกระจายงานแบบพลวัตอะซิงโครนัส (asynchronous dynamic load balancing) เพื่อกระจายงานและผลลัพธ์การประมวลผลระหว่างกันตามสภาวะการทำงานของแต่ละโหนด และเทคนิคการประมวลผลแบบกริด (grid computing) ที่คลัสเตอร์จะปรับจำนวนโหนดที่เหมาะสมตามขนาดของงาน (ไม่จำเป็นต้องทำงานทุกโหนด) ทำให้คลัสเตอร์สามารถทำได้หลาย ๆ งานพร้อมกัน และยังมีการสร้างสถาปัตยกรรมเปิดสำหรับคลัสเตอร์ที่ใช้ซอฟต์แวร์แบบอิสระและเปิดเผยต้นฉบับ (free and open source software) ชื่อ Beowulf โดยเริ่มจากคลัสเตอร์ต้นแบบที่มีชื่อเดียวกันคือ Beowulf cluster โดยระบบและสถาปัตยกรรมนี้จะทำงานอยู่บนระบบปฏิบัติการประเภทยูนิกซ์ (Unix-like operating system) เช่น Linux, Solaris และ BSD เป็นต้นและมีซอฟต์แวร์ไลบรารี (ที่พัฒนาไว้ก่อนแล้ว) เช่น PVM (Parallel Virtual Machine) และ

MPI (Massively Passing Interface) เป็นต้น สถาปัตยกรรม Beowulf ช่วยให้ผู้จัดการระบบสามารถสร้างคลัสเตอร์ที่สะดวกพร้อม ทั้งในการใช้งาน ปรับแต่งเป็นระบบที่พร้อมสำหรับการพัฒนาโปรแกรมที่ทำงานแบบขนาน (parallel programming)

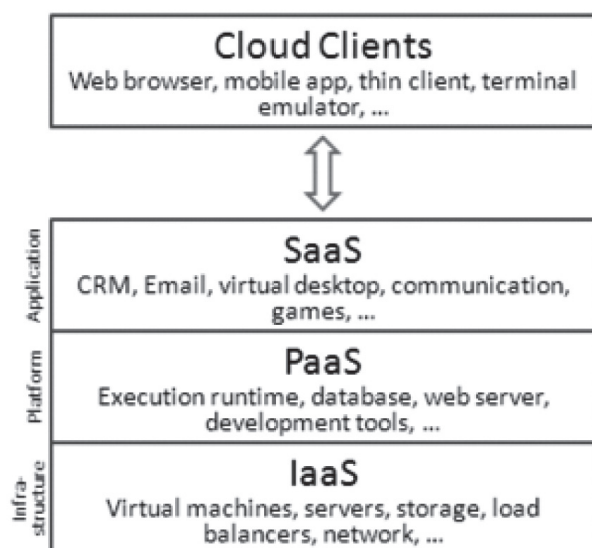
ในปัจจุบัน หน่วยประมวลผลกลางมีประสิทธิภาพสูงขึ้น นอกจากจะมีขนาดสถาปัตยกรรมที่กว้างขึ้น (64 บิต) แต่ละหน่วยประมวลผลกลางยังประกอบไปด้วยจำนวนหน่วยประมวลผลย่อยอยู่ภายในหลายหน่วยหรือที่เรียกว่าหลายคอร์ (multicore processor) และหน่วยประมวลผลร่วมทางกราฟิก (graphic coprocessor) ก็มีประสิทธิภาพสูง คอมพิวเตอร์ส่วนบุคคลตามบ้านเรือน หรือแม้แต่อุปกรณ์พกพาต่างก็ใช้หน่วยประมวลผลประเภทนี้แล้วแทบทั้งสิ้น ระบบปฏิบัติการและโปรแกรมยุคใหม่ก็ทำงานแบบขนาน อินเทอร์เน็ตก็ช่วยทำให้เครื่องแต่ละเครื่องเชื่อมต่อกันได้กว้างขวาง ยิ่งทำให้เหมาะกับการสร้างระบบการทำงานแบบขนานหรือกระจายไปทั่ว (distributing system) และซอฟต์แวร์หลายชนิดก็มีการทำงานแบบขนาน เมื่อถูกติดตั้งก็ทำให้เครื่องคอมพิวเตอร์เป็นโหนดหนึ่งในระบบการประมวลผลแบบกระจายนี้ เช่น BitTorrent ที่ใช้แบ่งปันไฟล์ โดยไม่ต้องมีผู้ให้บริการหลัก (server) หรือ Bitcoin Core ซึ่งเป็นระบบสนับสนุนธุรกรรมทางการเงินที่ใช้สกุลเงินทางอินเทอร์เน็ตที่ชื่อว่า Bitcoin เป็นต้น

แบบจำลองการให้บริการบนคลาวด์ (cloud service models)

องค์กรต่าง ๆ ที่มีระบบคอมพิวเตอร์แบบกระจายตัวหรือคลัสเตอร์ ต่างก็ต้องมีการลงทุนเพื่อสร้าง และมีค่าใช้จ่ายในการดำเนินงานและบำรุงรักษา หลายองค์กรจึงเปิดให้บริการแก่หน่วยงานภายนอกเข้ามาใช้งานระบบของตน ผู้ที่ต้องการใช้งานสามารถเช่าใช้ทรัพยากรของคลัสเตอร์ได้โดยอาจจะเสียค่าบริการให้แก่องค์กร และจำเป็นต้องมีโปรแกรมที่ออกแบบมาเฉพาะเพื่อนำไปรันบนคลัสเตอร์ (ใช้ไลบรารี PVM หรือ MPI และมีอัลกอริทึมที่เหมาะสมต่อการทำงานแบบขนาน) ตัวอย่างของโปรแกรมแบบขนานนี้ก็คือ เช่น โปรแกรมการจำลองการเกิดปฏิกิริยาในสารละลาย (จำลองการแตกตัวและรวมตัวของแต่ละอนุภาค) การทำนายสภาพอากาศ ฯลฯ ซึ่งองค์กรภายนอก

ที่มาขอใช้บริการของคลัสเตอร์ก็มักจะจำกัดแคววอยู่ที้องค์กรที่ทำงานที่เกี่ยวข้องกับด้านวิทยาศาสตร์เทคโนโลยีหรือวิชาการ

ต่อมาได้มีแนวคิดเพื่อให้ระบบกระจายตัวนี้ตอบสนองต่อการใช้งานไม่เฉพาะในด้านวิทยาศาสตร์และเทคโนโลยี แต่รวมถึงในด้านธุรกิจและอื่น ๆ ด้วย สถาบัน NIST (National Institute of Standards and Technology) ของประเทศสหรัฐอเมริกาจึงได้กำหนดสถาปัตยกรรมและมาตรฐานของแบบจำลองการให้บริการของระบบกระจายตัว (cloud service models) โดยเน้นการบริการเป็นหลัก (service oriented) และเริ่มมีการใช้คำว่าคลาวด์เพื่อใช้เรียกระบบกระจายตัวตามแบบสถาปัตยกรรมดังกล่าว แบบจำลองการให้บริการที่เป็นมาตรฐานนี้มีอยู่ 3 แบบจำลอง คือ Infrastructure as a Service (IaaS), Platform as a Service (PaaS) และ Software as a Service (SaaS)



ภาพ 1 แผนภาพแสดงแบบจำลองการให้บริการบนคลาวด์

Infrastructure as a Service (IaaS)

โดยทั่วไป องค์กรส่วนใหญ่จำเป็นต้องมีโครงสร้างพื้นฐานทางระบบคอมพิวเตอร์ ซึ่งอาจประกอบไปด้วยโครงข่ายและทรัพยากรทางคอมพิวเตอร์ (เช่น เซิร์ฟเวอร์เน็ตเวิร์คสวิตช์ ฯลฯ) องค์กรจำเป็นต้องลงทุนในการซื้อฮาร์ดแวร์และซอฟต์แวร์เพื่อสร้างโครงข่ายดังกล่าว และยังคงคอยบำรุงรักษา ปรับตั้ง และปรับเปลี่ยนอุปกรณ์เพื่อทดแทนในกรณีที่ล้มสมัยหรือชำรุด ซึ่งต้องใช้เวลา

เงินและทรัพยากรบุคคล IaaS เป็นการให้บริการที่เกิดจากการจำลองสร้างโครงข่ายและทรัพยากรเสมือน (virtual infrastructure) บนคลาวด์ โดยโครงข่ายเสมือนนี้สามารถจำลอง (simulate/virtualize) ให้มีโทโปโลยีทางเครือข่าย (network topology) ที่ประกอบด้วยเครื่องเสมือน (virtual machines) และอุปกรณ์เสมือน (virtual network devices) ตามที่ผู้ดูแลระบบ (network administrators) ในแต่ละองค์กรต้องการ เครื่องเสมือนเหล่านี้สามารถถูกกำหนดให้มีทรัพยากรอย่างที่ต้องการ เช่น ขนาดหน่วยความจำขนาดของดิสก์ หน่วยประมวลผล ฯลฯ โครงข่ายนี้จะเสมือนจริงทำให้ผู้ดูแลระบบและผู้ใช้ในองค์กรรู้สึกราวกับว่ากำลังใช้งานบนเครือข่ายจริงที่มีเครื่องเซิร์ฟเวอร์และอุปกรณ์เครือข่ายจริง (physical hardware) ผู้ดูแลระบบจะสามารถติดตั้งทั้งระบบปฏิบัติการ หรือโปรแกรมต่าง ๆ ลงในเครื่องคอมพิวเตอร์หรือเซิร์ฟเวอร์เสมือนเหล่านี้ได้โดยตรงเหมือนกับเครื่องจริง หรืออาจใช้ซอฟต์แวร์สร้างเครื่องเสมือนบนคอมพิวเตอร์ส่วนตัว (ที่มีทรัพยากรเหมือนกับเครื่องเสมือนบนเครือข่ายเสมือน) เพื่อติดตั้งระบบลงในเครื่องเสมือนนี้ก่อน แล้วจึงอัปโหลดดิสก์อิมเมจ (disk-image) ที่ได้จากเครื่องเสมือนส่วนตัวนี้ไปสู่เครื่องเสมือนในเครือข่ายเสมือนในภายหลังก็ได้ หรือจะใช้ดิสก์อิมเมจเดียวกันกับหลาย ๆ เครื่องเสมือนในเครือข่ายเสมือนโดยไม่ต้องเสียเวลาติดตั้งซ้ำในแต่ละเครื่อง และยังสามารถสร้างดิสก์อิมเมจที่แตกต่างกันสำหรับเครื่องเสมือนหนึ่ง ๆ เพื่อสลับเปลี่ยนใช้ได้ตามต้องการ ซอฟต์แวร์ที่สามารถสร้างเครื่องเสมือนเหล่านี้ก็มีทั้งที่เป็นซอฟต์แวร์เสรี เช่น OracleVirtualBox และเพื่อการค้า เช่น VMware หรือ Microsoft Hyper-V เป็นต้น

จากที่กล่าวมาจะเห็นได้ว่า องค์กรที่ต้องการใช้โครงข่ายเสมือนบนคลาวด์นี้สามารถติดต่อขอใช้บริการจากผู้ให้บริการคลาวด์ที่มีอยู่หลากหลายได้เช่น Amazon S3, Windows Azure, Google Compute Engine, IBM Smart Cloud Enterprise เป็นต้น องค์กรไม่จำเป็นต้องลงทุนซื้อฮาร์ดแวร์มาสร้างโครงข่ายจริงด้วยตนเอง ไม่ต้องหาสถานที่เพื่อจัดเก็บฮาร์ดแวร์ไม่ต้องบำรุงรักษา ปรับเปลี่ยนฮาร์ดแวร์ให้ทันสมัย ผู้ให้บริการย่อมต้องบำรุงและปรับปรุงฮาร์ดแวร์จริงของตนให้สามารถรองรับเครือข่ายเสมือนนี้ได้ แต่องค์กรก็จำเป็นต้องมีเครื่องไคลเอนต์ที่สามารถเชื่อมต่อเข้า

กับคลาวด์ (ผ่านทางอินเทอร์เน็ต) ได้อย่างรวดเร็ว มิฉะนั้น จะเกิดปัญหาคอขวดในการเข้าถึงทรัพยากรเสมือนเหล่านี้ นอกจากนี้ สำหรับองค์กรที่อาจจะมีข้อจำกัดในเรื่อง ความเร็วในการเข้าถึงอินเทอร์เน็ต หรือมีปัญหาอื่นทำให้ ไม่เหมาะที่จะรับบริการโครงข่ายเสมือนจากผู้ให้บริการ ก็สามารถสร้างคลาวด์เฉพาะในองค์กรจากโครงข่าย ท่อถิ่นในองค์กร และนำคลาวด์ที่สร้างขึ้นมาสร้างระบบ เสมือนนี้ได้ โดยไม่จำเป็นต้องเสียเงินจำนวนมาก เพราะมี ซอฟต์แวร์ที่สร้างคลาวด์ที่สามารถให้บริการแบบ IaaS นี้ เช่น OpenStack ของ OpenStack Foundation หรือ Ubuntu OpenStack (Rochwerger, et al., 2009) ของบริษัท Canonical (ผู้ผลิต Ubuntu Linux) เป็นต้น โครงข่ายและ ทรัพยากรเสมือนที่สร้างขึ้นโดยคลาวด์นี้ นอกจากจะมี ประสิทธิภาพใกล้เคียงกับโครงข่ายและทรัพยากรใน ฮาร์ดแวร์จริงแล้ว (ขึ้นอยู่กับฮาร์ดแวร์จริงที่มีอยู่ใน คลาวด์) เครื่องเสมือนที่สร้างขึ้นยังอาจมีประสิทธิภาพ ดีกว่าเครื่องจริงได้เสียอีก เช่น อาจมีความสามารถในการ ประมวลผลสูงกว่าเครื่องจริง มีความจุของหน่วยความจำ หรือฮาร์ดดิสก์มากกว่าเครื่องจริงก็ได้

Platform as a Service (PaaS)

สำหรับองค์กรที่มีเป้าหมายในการพัฒนา ซอฟต์แวร์ จำเป็นต้องมีเครื่องเป้าหมายที่จะพัฒนา โปรแกรมให้ใช้บนเครื่องนั้น ๆ แต่เดิมองค์กรจะกำหนด เครื่องหรือแพลตฟอร์มที่ใช้ในการผลิตงาน (production system) เพื่อใช้ในการพัฒนาติดตั้งและทดสอบโปรแกรม ที่พัฒนาขึ้น เครื่องดังกล่าวจะต้องมีเสถียรภาพ ไม่มีการลง ซอฟต์แวร์อื่นเกินจำเป็น และหากจะมีการปรับปรุง ซอฟต์แวร์หรือฮาร์ดแวร์ต้องได้รับการยินยอมหรือตกลง จากทีมพัฒนาเสียก่อน เพื่อป้องกันไม่ให้เกิดจาก ซอฟต์แวร์อื่นหรือฮาร์ดแวร์มารบกวน (แม้แต่โปรแกรม แอนตี้ไวรัสก็มักไม่ถูกติดตั้ง) แพลตฟอร์มที่ใช้ในการผลิต ซอฟต์แวร์นี้จะรวมทั้งฮาร์ดแวร์และซอฟต์แวร์ที่ใช้ เช่น ชนิดขั้นต่ำของหน่วยประมวลผลกลาง ขนาดหน่วยความจำ ระบบปฏิบัติการที่ใช้ ฮาร์ดแวร์และหรือซอฟต์แวร์อื่นที่ เกี่ยวข้อง เป็นต้น ซึ่งหลายแพลตฟอร์มสามารถสร้างขึ้น ได้จากการใช้ซอฟต์แวร์สร้างเครื่องเสมือน แต่ PaaS เป็น รูปแบบการบริการที่ผู้ให้บริการจะจำลองแพลตฟอร์ม

ให้กับผู้ใช้บริการสร้างโปรแกรมสำหรับคลาวด์ (cloud software) โดยแพลตฟอร์มนี้จะเป็นคลาวด์แพลตฟอร์ม (cloud platform) แบบเสมือน ทำให้นักพัฒนาไม่จำเป็นต้องทราบโครงสร้างทางฮาร์ดแวร์จริง ๆ บนคลาวด์ ที่ตนเองกำลังพัฒนานั้นเป็นอย่างไร มีจำนวนโหนดมากน้อย แค่ไหน และเมื่อมีการปรับปรุงหรือเปลี่ยนแปลงทาง ฮาร์ดแวร์ แพลตฟอร์มก็จะสามารถใช้ประสิทธิภาพที่ เพิ่มขึ้น แต่ยังคงเสถียรภาพที่เหมาะสมต่อการพัฒนาได้ เช่นเดิม

ผู้ให้บริการ PaaS จะสร้างชุดเครื่องมือ (toolkit) ซอฟต์แวร์ไลบรารี และกำหนดมาตรฐานในการใช้งาน แพลตฟอร์มเหล่านี้ การเขียนโปรแกรมแบบขนานแบบ ดังเดิมที่ใช้ซอฟต์แวร์ไลบรารี PVM หรือ MPI ผู้ใช้สามารถ ควบคุมจำนวนโหนดที่เรียกใช้งาน ปริมาณและลำดับการ เข้าถึงของข้อมูลหรืองานที่กำหนดให้แต่ละโหนด แต่ ในคลาวด์แพลตฟอร์ม มักจะไม่สามารถกำหนดลำดับ ของงานหรือข้อมูลที่กระจายให้กับแต่ละโหนดได้

เทคนิคการเขียนโปรแกรมอย่างเช่น Map Reduce ซึ่งแบ่งออกกลุ่มของโปรเซสออกเป็น 2 กลุ่ม คือ Map และ Reduce กล่าวโดยสั้น Map จะทำการค้นหาและประมวล ข้อมูลที่ตรงตามเงื่อนไขที่กำหนดมาพักไว้ Reduce จะนำ ข้อมูลที่พักไว้มาประมวลผลต่อให้ได้คำตอบ (มีส่วน คล้ายกับ stored procedure ในซอฟต์แวร์ฐานข้อมูลทั่วไป) ข้อมูลมักจะถูกเก็บไว้ก่อนแล้วเป็นจำนวนมากในไฟล์ (หลายไฟล์) หรือในฐานข้อมูลก็ได้ ผู้พัฒนาเพียงเขียน โปรแกรมว่า Map และ Reduce มีการประมวลผลกับข้อมูล เดียว ๆ ที่เจออย่างไร เมื่อเริ่มการทำงาน คลาวด์เอนจินจะ กระจายโปรเซส Map ไปตามโหนด และส่งข้อมูลกระจาย ไปตามโหนดโดยไม่มีลำดับที่แน่นอน ข้อมูลจะได้รับการ ประมวลผลจากโปรเซส Map และถูกพักเก็บไว้เมื่อข้อมูล ถูกส่งและประมวลผลจนครบหมดแล้ว คลาวด์เอนจิน จะกระจายโปรเซส Reduce ไปตามโหนด และส่งข้อมูลที่ พักเก็บไว้ไปให้กับแต่ละโหนดโดยไม่เรียงลำดับ เพื่อ ประมวลผลต่อให้ได้เป็นผลลัพธ์ (และเก็บผลลัพธ์นั้น ในไฟล์หรือฐานข้อมูล)

ตัวอย่างของผู้ให้บริการแบบ PaaS ได้แก่ Red Hat OpenShift, Google App Engine, Windows Azure Cloud Services, Amazon AWS และสำหรับผู้ที่ต้องการสร้าง

คลาวด์แพลตฟอร์มขึ้นเอง ก็สามารถใช้ซอฟต์แวร์เสรี เช่น Apache Hadoop ได้

Software as a Service (SaaS)

SaaS เป็นบริการที่ผู้ให้บริการจะพัฒนาคลาวด์ซอฟต์แวร์ เพื่อตอบสนองความต้องการของผู้ใช้ โดยคลาวด์ซอฟต์แวร์จะต่างจากซอฟต์แวร์อื่น ตรงที่สามารถใช้ประโยชน์จากการประมวลผลบนคลาวด์มาใช้ในการใช้ประโยชน์จากการประมวลผลบนคลาวด์มาใช้ในซอฟต์แวร์ เช่น อาจ会有ความสามารถในการประมวลผลที่มีประสิทธิภาพสูง หรือสามารถจัดเก็บข้อมูลขนาดใหญ่ และมีความปลอดภัย หรือเข้าถึงข้อมูลที่หลากหลาย หรือการได้ข้อมูลหรือผลแบบเวลาจริง ฯลฯ เป็นต้น

คลาวด์ซอฟต์แวร์มีอยู่หลากหลายและนับวันจะมีมากยิ่งขึ้น บางบริการก็ไม่ได้คิดค่าบริการ แต่บางบริการก็คิดหรือเปิดให้เฉพาะกับสมาชิกผู้ใช้อินเทอร์เน็ตในปัจจุบันต่างก็เคยใช้บริการคลาวด์ซอฟต์แวร์แล้วแต่อาจจะไม่รู้ตัว เช่น Google Search Engine, Gmail, Facebook, DropBox เป็นต้น โดยทั่วไปผู้ใช้บริการนี้มักจะเป็นผู้ใช้ทั่วไป แต่ก็มีผู้ใช้ในระดับองค์กรเช่นกัน เช่น Gmail for Business ที่ให้บริการอีเมลสำหรับลูกค้าระดับองค์กร โดยบุคลากรในองค์กรสามารถใช้ที่อยู่อีเมลประจำองค์กรของตน เช่น nikorn@eau.ac.th เพื่อล็อกอินเข้าสู่กล่องรับจดหมายของตนผ่านทางหน้าล็อกอินปกติของ Gmail ได้

หากไม่มีคลาวด์ซอฟต์แวร์ต่าง ๆ ก็อาจจะพัฒนาขึ้นได้ แต่จะมีประสิทธิภาพไม่เท่ากับคลาวด์ซอฟต์แวร์ เช่น เซิร์ฟเวอร์ที่ไม่ใช้คลาวด์ อาจจะไม่สามารถค้นข้อมูลมหาศาลได้อย่างรวดเร็ว อาจต้องจำกัดจำนวนผู้ใช้ในแต่ละช่วงเวลา ไม่มีการแนะนำคำค้นแบบเวลาจริง ไม่มีการแนะนำคำแก้คำผิด เป็นต้น

IoT กับ Cloud Computing

กลับมาดูตัวอย่างอุปกรณ์นำทางที่มีการเชื่อมต่ออินเทอร์เน็ตที่ได้กล่าวไว้ในตอนแรกว่าจะช่วยให้สามารถแสดงความหนาแน่นและหาเส้นทางหลีกเลี่ยงการจราจรที่ติดขัดได้ตามเวลาจริงนั้น ถ้าอุปกรณ์นี้ไม่ได้เชื่อมต่ออยู่กับคลาวด์ จะไม่สามารถทำงานอย่างที่เราได้เคยทำได้เพียงปรับปรุงเฟิร์มแวร์ใหม่ หรือปรับปรุงแผนที่

จากเว็บไซต์ของผู้ผลิตเท่านั้น เพราะการจะได้ข้อมูลจริงทางด้านการจราจรนั้น จำเป็นต้องมีตัวตรวจวัดข้อมูลการจราจรอยู่ทั่วไปที่สามารถส่งข้อมูลให้กับคลาวด์เพื่อประมวลผลหาความหนาแน่นได้

พิจารณาการทำงานของระบบนำทางใน Google Map บนโทรศัพท์มือถือ ซึ่งเป็นคลาวด์ซอฟต์แวร์ที่มีความสามารถในการแสดงความหนาแน่นของการจราจรดังกล่าว เริ่มจากที่โทรศัพท์มือถือแต่ละเครื่องจะสามารถหาพิกัด (geolocation) ได้จากข้อมูลจากดาวเทียม หรือไวไฟ สอตสปอต หรือจากลงทะเบียนเข้ากับเครือข่ายโทรศัพท์เคลื่อนที่ผ่านเซลล์ไซต์ (cell site) ถ้าโทรศัพท์เคลื่อนที่แต่ละเครื่องที่อยู่บนถนนเชื่อมต่อเข้ากับอินเทอร์เน็ตได้ หรือผู้ให้บริการเครือข่ายโทรศัพท์เคลื่อนที่ส่งข้อมูลของเซลล์ไซต์ให้กับคลาวด์ของกูเกิล บรรดาโทรศัพท์เคลื่อนที่ทั้งหลายก็จะเป็นเสมือนตัวตรวจวัดสภาพการจราจร ช่วยให้คลาวด์ของกูเกิลคำนวณหาความหนาแน่นของการจราจร และสามารถส่งข้อมูลนี้มาแสดงผลหรือคำนวณต่อใน Google Map ที่อยู่ในโทรศัพท์มือถือได้

และเพื่อให้เห็นชัดว่าการประมวลผลแบบคลาวด์นั้นช่วยเสริมประสิทธิภาพให้กับ IoT อย่างไร ให้เราพิจารณาตัวอย่างของกล้องวงจรปิดสำหรับระบบรักษาความปลอดภัยภายในบ้านระหว่างกล้องแบบไอพีที่เป็น IoT และแบบคลาวด์ คุณสมบัติและลักษณะเด่นของกล้องวงจรปิดนั้นมีหลากหลายมาก เช่น การบันทึกภาพเป็นแบบอนาล็อกหรือดิจิทัล ความละเอียดของภาพความสามารถในการถ่ายในที่แสงน้อย มีแสงอินฟราเรด และเลนส์เพื่อถ่ายในที่มืด มีเลนส์สำหรับบันทึกภาพมุมกว้างหรือพิกซอล ปรับโฟกัสอัตโนมัติ สามารถสายได้ตามจังหวะหรือตามต้องการ สามารถตรวจจับความเคลื่อนไหวแล้วค่อยบันทึกหรือเปิดสัญญาณกันขโมย เป็นต้น

ปัญหาของการใช้กล้องวงจรปิดที่สร้างความยากลำบากให้กับผู้ใช้งานมากปัญหาหนึ่งก็คือ ปัญหาในการควบคุมหรือดูภาพจากกล้องวงจรปิดขณะไม่อยู่ในบ้าน แม้ว่ากล้องจะต่อเข้ากับอินเทอร์เน็ต แต่ก็มักจะสามารถควบคุมและเข้าถึงได้จากเครื่องที่ต่ออยู่กับเครือข่ายในบ้านเท่านั้น ปัญหานี้เกิดจากการที่หมายเลขไอพีแอดเรสของโมเด็มตามบ้านนั้นไม่ใช่หมายเลขถาวร เพราะทุกครั้งที่โมเด็ม

เชื่อมต่อเข้าสู่อินเทอร์เน็ตผ่านทางผู้ให้บริการอินเทอร์เน็ต (internet service provider) ผู้ให้บริการจะจัดสรรหมายเลขที่ว่างอยู่ให้กับโมเด็มที่ร้องขอ ซึ่งอาจเป็นหมายเลขเดิมที่เคยใช้หรือหมายเลขใหม่ก็ได้ เมื่อผู้ใช้อยู่นอกบ้านไม่สามารถรู้หมายเลขไอพีแอดเดรสของโมเด็มที่บ้าน ก็ทำให้ไม่สามารถเข้าถึงเครือข่าย กล้อง หรือเซิร์ฟเวอร์บันทึกภาพที่บ้านได้

วิธีแก้ปัญหาโดยทั่วไปคือ ใช้บริการ DDNS (Dynamic Domain Name System) เพื่อระบุหมายเลขไอพีแอดเดรสปัจจุบันของโมเด็มเข้ากับชื่อโดเมนที่คงที่และเขียนไว้ เช่น ถ้าผู้ให้บริการ DDNS เป็น ip.org และคงที่เขียนชื่อโดเมนเป็น mycamera.ip.org โมเด็มหรือเซิร์ฟเวอร์บันทึกภาพจะส่งหมายเลขไอพีและชื่อโดเมนของโมเด็มไปที่ ip.org เมื่อผู้ใช้อยู่นอกบ้านและขอติดต่อกล้องผ่านทางโดเมน mycamera.org ก็จะได้ไอพีแอดเดรสที่ถูกต้องและเข้าถึงกล้องได้ในที่สุด ภาพที่ต้องการจะถูกส่งตรงจากกล้องมายังอุปกรณ์ของผู้ใช้ ซึ่งผู้ให้บริการ DDNS นี้อาจเป็นผู้บริการทั่วไปหรือเป็นผู้ผลิตกล้องวงจรปิดนั้น ๆ ก็ได้

กล้องแบบคลาวด์จะแตกต่างออกไปคือ กล้องจะส่งภาพไปบันทึกลงในคลาวด์และติดต่อขอคำสั่งในการควบคุมกล้องจากคลาวด์ (ในกรณีที่ผู้ใช้ส่งคำสั่งควบคุมกล้องมาที่คลาวด์) เมื่อผู้ใช้อยู่นอกบ้าน ก็สามารถเข้าถึงคลาวด์นั้นและดูภาพที่บันทึกได้ และถ้าดูภาพสดจากกล้องตัวเดียวกันจากหลายอุปกรณ์พร้อมกัน ก็ไม่ทำให้แบนด์วิธการเชื่อมต่อที่บ้านเพิ่มสูงขึ้นมากไปกว่าการดูด้วยอุปกรณ์เดียวแต่อย่างใด

ในกรณีที่บันทึกกล้องเซิร์ฟเวอร์ก็เหมาะกับบ้านที่มีกล้องเป็นจำนวนมาก และแบนด์วิธในการเชื่อมต่อ

อินเทอร์เน็ตไม่เพียงพอจะส่งภาพจากทุกกล้องไปยังคลาวด์ได้แต่ภาพที่บันทึกไว้และตัวเซิร์ฟเวอร์เองก็อาจเสี่ยงต่อการสูญหายหรือถูกทำลายการบันทึกลงในคลาวด์จะป้องกันการสูญหายหรือทำลายภาพได้ดีกว่า และแม้มีกล้องหลายตัว ถ้ามีการบันทึกเฉพาะถ้ามีการเคลื่อนไหวในภาพตามบริเวณที่กำหนดก็จะลดข้อมูลและแบนด์วิธในการเชื่อมต่อเข้ากับคลาวด์ลงได้

บทสรุป

การประมวลผลบนคลาวด์มีความสัมพันธ์และมีประโยชน์กับ IoT ช่วยให้อุปกรณ์ที่เป็น IoT สามารถเข้าถึงศักยภาพที่แท้จริงของอินเทอร์เน็ตได้ ไม่ว่าจะเป็นการประมวลผลที่มีความเร็วสูงการได้มาซึ่งข้อมูลที่หลากหลายและตามเวลาจริง การรับส่งข้อมูลจำนวนมาก การรักษาความปลอดภัยของข้อมูล ฯลฯ ซึ่งตอบสนองต่อจุดประสงค์และแนวคิดของ IoT ในการสร้างอุปกรณ์ชาญฉลาด

กล่าวได้ว่า การประมวลผลบนคลาวด์นั้นเป็นโครงสร้างพื้นฐานที่สำคัญยิ่งสำหรับ IoT

ทิศทางในอนาคต

ในอนาคตรูปแบบของการประมวลผลบนคลาวด์จะเป็นไปอย่างทั่วถึงมากขึ้น จะมีโมเดลในการให้บริการของคลาวด์เพิ่มขึ้นอีกมากเพื่อตอบสนองความต้องการ โดยเฉพาะอย่างยิ่งในด้านธุรกิจ พาณิชย์ และบันเทิงตามแนวคิดที่ว่า ทุกอย่างเป็นบริการ หรือ XaaS (Everything as a Service) ซึ่งจะมีผลต่อ IoT และอุปกรณ์ชาญฉลาดที่ฉลาดยิ่งขึ้นและเทคโนโลยีที่ก้าวหน้าจะทำให้อุปกรณ์เหล่านี้ไม่เพียงเป็นเพียงไคลเอนต์ที่มาใช้บริการ แต่จะกลายเป็นโหนดหนึ่งที่ร่วมประมวลผลบนคลาวด์ด้วย



References

- Armbrust, M., Fox, A., Griffith, R., Joseph, D. A., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., & Zaharia, M. (2010). *A view of cloud computing*. Retrieved from <http://cacm.acm.org/magazines/2010/4/81493-a-view-of-cloud-computing/fulltext>
- Millard, C. (2013). *Cloud computing law*. Oxford: Oxford University.
- Qusay, H. (2011). *Demystifying cloud computing*. Retrieved from https://www.researchgate.net/publication/215795101_Demystifying_Cloud_Computing
- Rochwerger, B., Breitgand, D., Levy, E., Galis, A., Nagin, K., Llorente, I. M., Montero, R., Wolfsthal, Y., Elmroth, E., Caceres, J., Ben-Yehuda, M., Emmerich, W., & Galan, F. (2009). *The reservoir model and architecture for open federated cloud computing*. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.330.3880&rep=rep1&type=pdf>
- Tung-Hui, H. (2015). *A prehistory of the cloud*. Massachusetts: MIT.

