Scaling UBER

โดย

นาย ฐิติ ชื่นบุบผา

5809450025

รายงานนี้เป็นส่วนหนึ่งของวิชา

Operating System 2

CS447

คณะวิทยาศาสตร์และเทคโนโลยี

มหาวิทยาลัยธรรมศาสตร์

ปีการศึกษา 2560

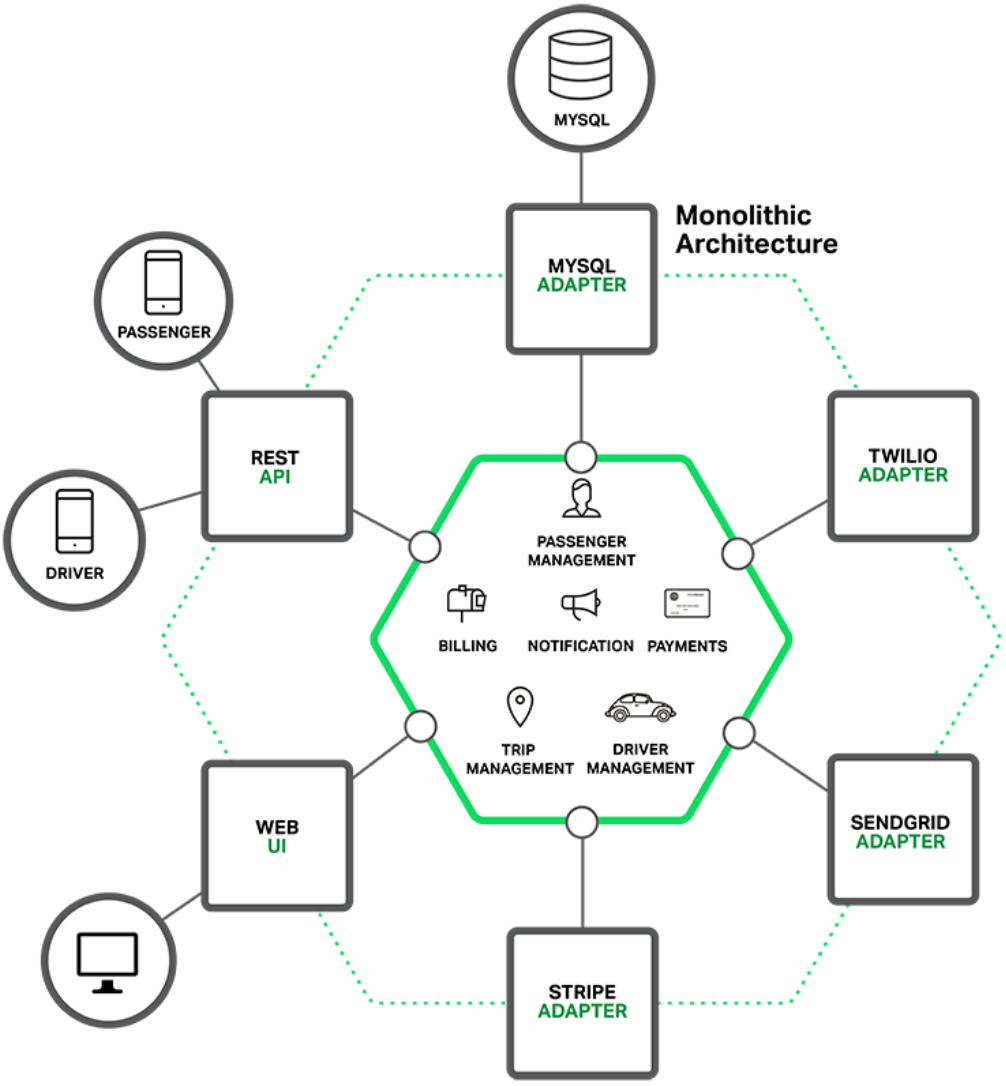
Scaling UBER

บริษัทดังกล่าวให้บริการอะไร

Uber เป็นบริษัทเครือข่ายคมนาคม ให้บริการการรับคำขอโดยสารจากผู้โดยสาย ไปยังคนขับของตน เพื่อการให้บริการคมนาคม หรือ ให้บริการผู้ที่มีรถยนต์และต้องการหาผู้โดยสายเพื่อสร้างรายได้พิเศษได้เช่นเดียวกัน โดยมีบริการต่างๆ ที่อาศัยเทคโนโลยีในปัจจุบันมาใช้ในการพัฒนา เช่น การค้นหาตำแหน่งของสถานที่ ผู้โดยสาร หรือ รถยนต์ที่ให้บริการ และยังใช้เทคโนโลยีในการค้นหาเส้นทางในการเดินทาง การคำนวณเวลาในการเดินทาง รวมไปถึงการประมาณค่าโดยสารในการเดินทาง

ปัญหาด้านการ **Scalability** ในการให้บริการ

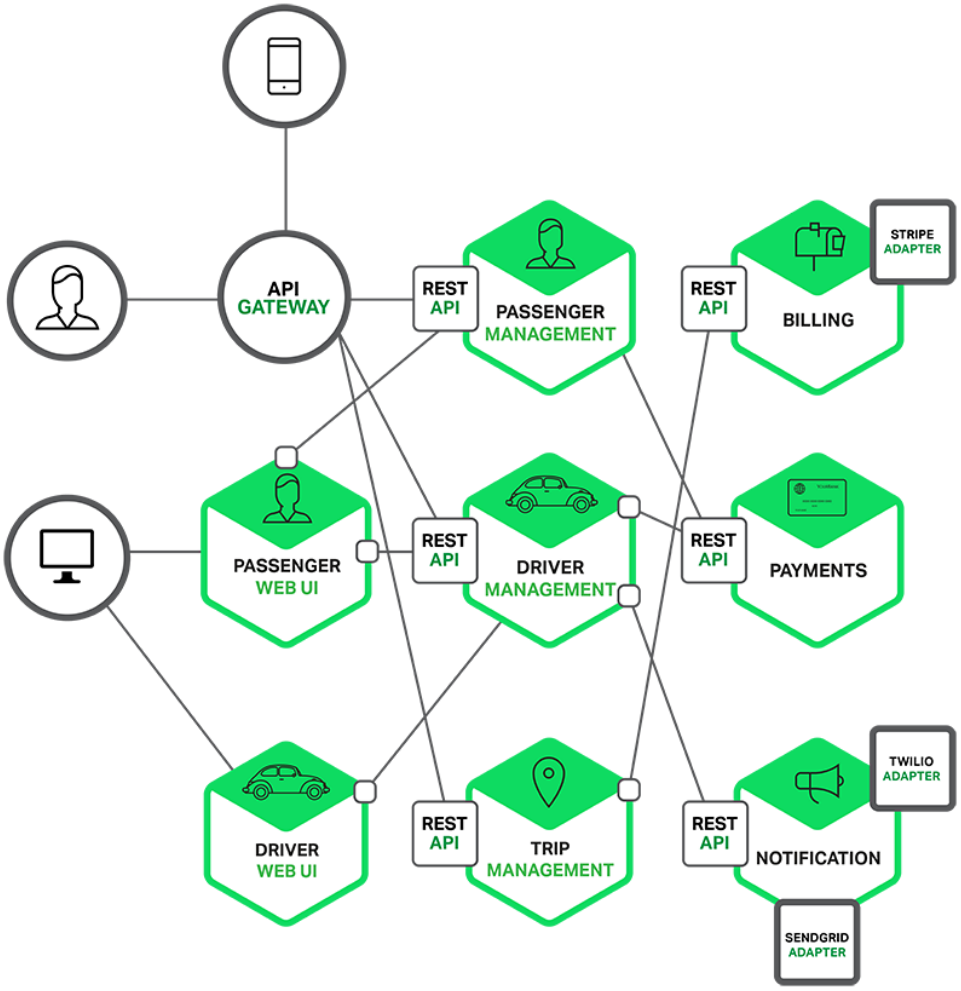
เนื่องจาก Uber เป็นบริษัทที่มีอัตราการเติบโตของวิศวกรที่สูง และมีอัตราการเพิ่ม service ต่างๆของระบบที่สูงเช่นเดียวกัน ทำให้การทำงานในการพัฒนา service ต่างๆ ให้สอดคล้องและไปในทิศทางเดียวกันนั้นจึงเป็นเรื่องที่ทำได้ยาก เนื่องจากระบบใช้ Architecture แบบ Monolithic Architecture ทำให้ส่วนของงานมีขนาดใหญ่และซับซ้อน การจะเพิ่ม service ขึ้นมาใหม่ จะต้องคำนึงถึงระบบเก่าที่ออกแบบเอาไว้ และการเปลี่ยนแปลง service ใด ๆ อาจส่งผลกระทบถึงระบบโดยรวมได้ อีกทั้งการ deploy นั้นใช้เวลานานเนื่องจากระบบนั้นมีขนาดใหญ่ และการแก้ไขเพียงบางส่วนของระบบจำเป็นจะต้องทำการ deploy ใหม่ทั้งระบบ



สถาปัตยกรรมและเทคนิควิธีที่ใช้ในการแก้ปัญหา

1.)Microservices

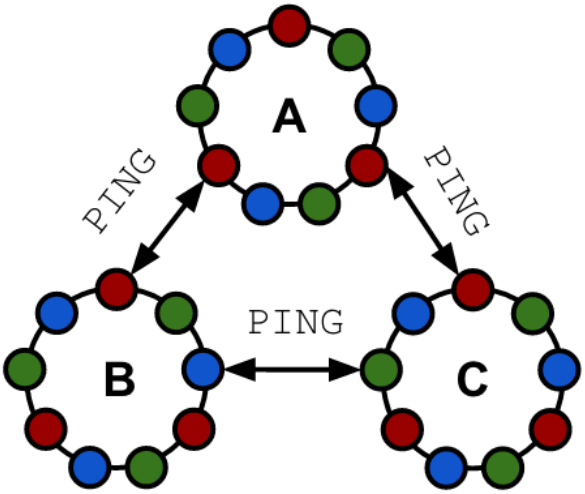
Microservices เป็นสถาปัตยกรรมที่ออกแบบโดยการแบ่งระบบใหญ่ออกเป็น service ที่มีขนาดเล็กเป็นจำนวนมาก โดยแต่ละ services นั้นจะเป็นอิสระต่อกันมีฐานข้อมูลเป็นของตนเอง สามารถ deploy แยกจากกันแต่ยังสามารถทำงานร่วมกันได้ และมีทำงานเพียงจุดประสงค์เดียว โดยการเพิ่มหรือเปลี่ยนแปลงการทำงานของ services ใดๆนั้น จะไม่ส่งผลกระทบต่อ services อื่นๆ (Immutable) และการพัฒนา service ใหม่ๆ จะต้องไม่แก้ไขหรือเปลี่ยนแปลงการทำงานของ service ที่มีอยู่ อีกทั้ง microservices ยังสนับสนุนการใช้ tools ที่เหมาะสมกับ services นั้นๆ โดยไม่ต้องสนใจว่าระบบดังกล่าวจะรองรับหรือไม่ เนื่องจากเกิดการทำงานที่แยกกันอย่างสมบูรณ์



2.) Ringpop

Ringpop คือ library ที่ใช้สำหรับช่วยในการติดต่อสื่อสารในการทำงานระหว่าง instance ที่เป็นอิสระต่อกัน Uber ได้พัฒนา Ringpop ขึ้นมาเพื่อใช้สำหรับทำการสื่อสารในการทำงานระหว่าง services ต่างๆที่มาจากการใช้ Microservices Architecture และยังช่วยในการ scaling service ต่างๆที่ถูกแบ่งย่อยให้มีประสิทธิภาพมากยิ่งขึ้น อีกทั้งยังจำเป็นในการจัดการการเพิ่มหรือลด จำนวน service ภายใน hash ring เพื่อให้ระบบสามารถทำงานได้อย่างถูกต้อง

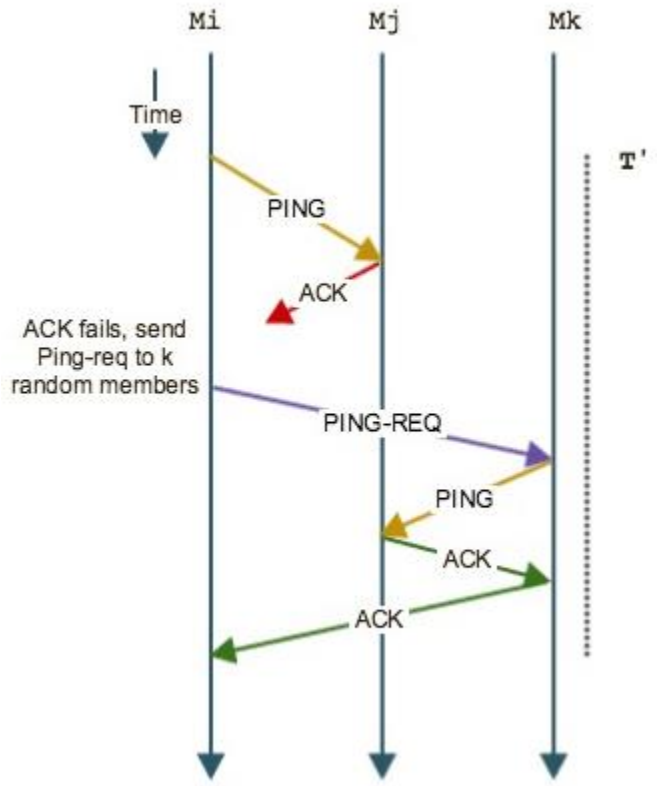
การทำงานของ Ringpop โดยทำการติดตั้ง Ringpop libraly ให้กับทุกๆ service ที่ deploy ระบบจะทำการสร้างช่องทางสื่อสาร โดย uber เลือกใช้ SWIM protocol ในการสื่อสารระหว่าง service และเมื่อทำการสร้างช่องทางการสื่อสารเสร็จเรียบร้อย ระบบจะทำการสร้าง hash ring ซึ่งเป็น group ของ communicate และทุกๆ service จะมีสถานะของทุกๆ service เก็บเอาไว้ และในการทำงานระหว่าง service นั้น Ringpop จะต้องทำการ forward request ไปยัง instance ที่ถูกต้อง อีกทั้ง Ringpop ยังต้องทำการส่งต่อ request ไปยัง node ที่มี traffic ที่ดีอีกด้วย เพื่อช่วยในเรื่องของการ scalability



3.) SWIM protocol

คือ protocol ที่ใช้ในการบริหารจัดการสมาชิกของกลุ่ม process และทำ failure detection ในแบบกระจาย ไม่มีจุดกลางที่ทำหน้าที่เป็นจุดเชื่อมต่อ ประกอบไปด้วยสองส่วนประกอบหลักคือ failure detector ใช้ในการตรวจหาสมาชิกที่หยุดทำงาน และ dissemination คือการกระจายข้อมูลออกไปยังสมาชิกในกลุ่ม Uber ใช้ SWIM protocol ในการตรวจหา service ที่หยุดให้บริการ และทำการกระจายข่าวออกไปยังทุกๆ service เพื่อไม่ให้ทำการเรียกใช้ service ที่ไม่พร้อมให้บริการ และเมื่อ service กลับมาให้บริการตามปกติก็จะทำการกระจายข่าวออกไปยังทุกๆ service เช่นเดียวกัน

โดยวิธีการทำงานคือ ในแต่ละช่วงเวลา node ต่างๆ จะทำการ ping ไปยัง node อื่นๆเพื่อเช็คสถานะอยู่เสมอ หาก node ที่ ping ไปนั้น ไม่ ack กลับมา node นั้นจะทำการ ping request node นั้นๆ ผ่านทาง node อื่นๆ เพื่อเช็คให้แน่ใจว่า node นั้นไม่พร้อมใช้งานจริง ไม่ใช่การ loss packet หาก node อื่นๆไม่ได้รับการตอบสนองเช่นเดียวกัน ก็จะประกาศให้ทุก node ทราบว่า node นั้นไม่พร้อมใช้งาน



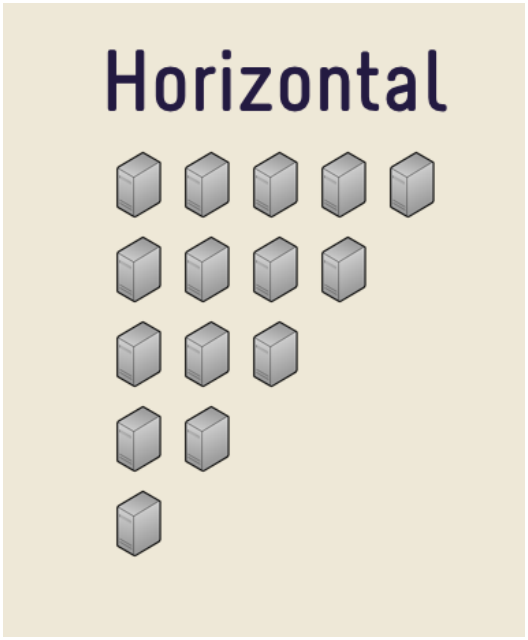
ผลที่ได้เมื่อใช้เทคนิควิธีดังกล่าว

- สามารถพัฒนา service ต่างๆได้อย่างต่อเนื่อง เนื่องจากการออกแบบ Architecture แบบ Microservices นั้นทำให้ service ต่างๆมีอิสระต่อกัน สามารถเพิ่ม service ได้โดยไม่ต้องทำการ deploy ใหม่
- มีความคล่องตัวในการพัฒนา ผู้พัฒนาสนใจเฉพาะ service ที่ตนเองพัฒนา ทำให้สามารถสร้าง service ใหม่ๆ ได้พร้อมกัน โดยที่ไม่ต้องรอ
- Tools ที่ใช้ในการ implement service ต่างๆ มีความหลากหลายและเหมาะสมในแต่ละงาน
- สามารถพัฒนา หรือแก้ไขเฉพาะบาง service ได้โดยง่าย โดยไม่ต้องไปยุ่งกับทั้งระบบ และไม่เสี่ยงต่อการสร้างความเสียหายต่อทั้งระบบ
- Failure detection สามารถตรวจสอบ service ที่ไม่พร้อมใช้บริการได้
- ระบบมีความ Availability หาก service ใดพัง service อื่นก็ทำงานต่อได้โดยไม่พังทั้งระบบ
- Fault tolerant หากมี service ใดไม่พร้อมบริการ สามารถส่งต่อการทำงานไปยังอีก instance ของ service เพื่อทำงานต่อได้
- ระบบมีความ Scalability สามารถเพิ่ม instance ของ service ต่างๆ หรือเพิ่มเฉพาะ service ที่ถูกใช้งานมากเป็นพิเศษได้

ความสัมพันธ์กับเนื้อหาที่เรียน

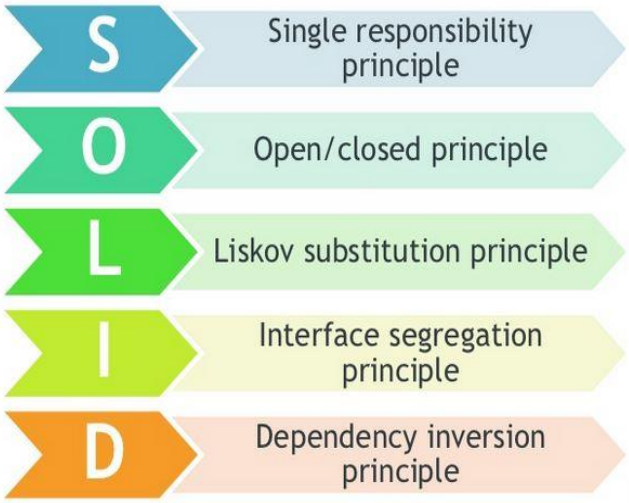
1.)Horizontal Scaling

มีการทำ Horizontal Scaling ในการเพิ่มจำนวน instance ของ service ต่างๆ



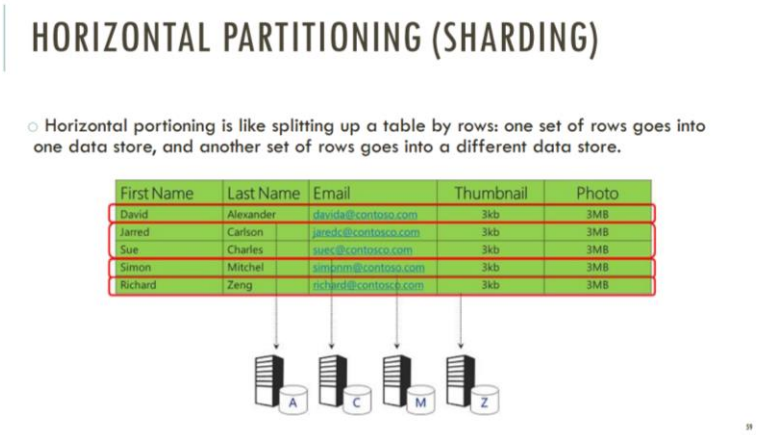
2.)SOLID Principle

Microservices are SOLID



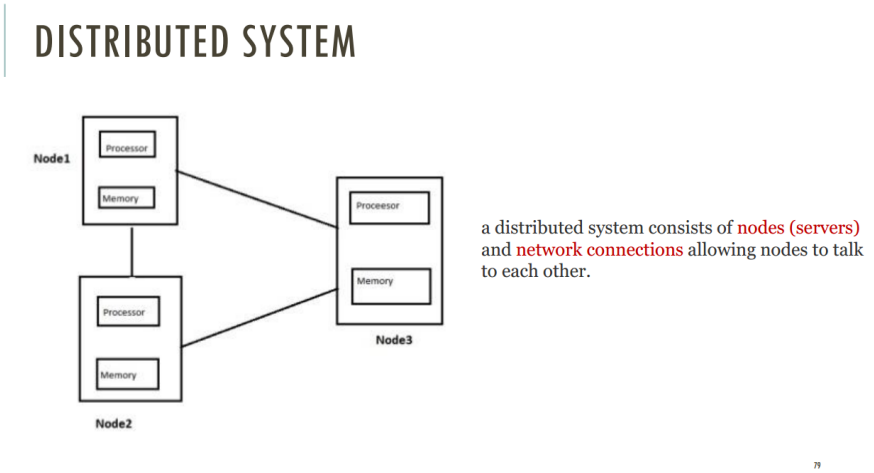
3.)Horizontal Partitioning(Shading)

แต่ละ service มี data base เป็นของตัวเอง ทำให้มี data ที่ต่างกัน



4.)Distributed System

Microsystems เป็น Distributed System



อ้างอิง

- Scaling Uber
  https://www.infoq.com/presentations/uber-scalability-arch
- Lessons Learned From Scaling Uber To 2000 Engineers, 1000 Services, And 8000 Git Repositories
  http://highscalability.com/blog/2016/10/12/lessons-learned-from-scaling-uber-to-2000-engineers-1000-ser.html
- ทำความรู้จักกับ Microservices
  https://www.techtalkthai.com/introduction-to-microservices-architecture/
- Ringpop
  https://eng.uber.com/intro-to-ringpop/
- การออกแบบและพัฒนาเฟรมเวิร์คสำหรับระบบไมโครเซอร์วิสแบบกระจาย
  http://ethesisarchive.library.tu.ac.th/thesis/2015/TU_2015_5409035192_4495_2642.pdf
- Ringpop Doc
  https://ringpop.readthedocs.io/en/latest/
- Uber change architecture to microservices
  https://eng.uber.com/building-tincup/