Written by Dr. Thapanapong Rukkanchanunt

11 Web Application Development I

OUTLINE

- Web Application Development
- Front-end Development
- Back-end Development
- Best Practice
- Performance Testing

Web Application Development

- Web Application Development หรือการพัฒนาเว็บแอพพลิเคชั่น คือขั้นตอนการสร้างโปรแกรมแอพพลิเคชั่นที่อาศัยอยู่ในเซิฟเวอร์ภายนอกแล้วถูกส่งต่อมายังอุปกรณ์ของผู้ใช้งานผ่านเครือข่ายอินเทอร์เน็ต
- เว็บแอพพลิเคชั่น (เว็บแอพ) ไม่จำเป็นต้องดาว์นโหลด สามารถเข้าถึงได้ผ่านเครือข่าย
- ผู้ใช้งานสามารถเข้าถึงเว็บแอพผ่านทางเว็บบราวเซอร์ เช่น Google Chrome, Safari หรือ Mozilla Firefox
- เว็บแอพส่วนมากจะถูกเขียนได้ภาษา JavaScript, Cascading Style Sheets และ HTML5

Front-end

- การพัฒนาเว็บแอพพลิเคชั่นมักจะมีวงจรการพัฒนาในระยะสั้น (Short Development Life-Cycle) นำโดยทีมพัฒนาขนาดเล็ก
- Front-end จะถูกพัฒนาด้วยการเขียนโปรแกรมฝั่งไคลเอนต์ (Client-side Programming)
- ภาษาที่ใช้ในการเขียน Front-end ได้แก่ HTML, CSS และ JavaScript

Back-end

- การเขียนโปรแกรมฝั่งเซิฟเวอร์ (Server-side Programming) จะช่วยให้การเขียนโปรแกรมฝั่งไคลเอนต์ทำงานได้ และสร้าง Script ให้เว็บแอพเรียกใช้งานได้
- บางครั้งเราเรียกส่วนของการพัฒนานี้ว่า Back-end
- การเขียน Script สามารถทำได้หลายภาษาเช่น Ruby, Java และ Python
- Script เหล่านี้จะกลายไปเป็น Interface ในมุมมองของผู้ใช้ นั่นแสดงว่าผู้ใช้จะไม่เห็น Source Code ของ Back-end
- หากเว็บแอพมีการจัดเก็บข้อมูลในฐานข้อมูล ก็จะทำผ่าน Back-end ด้วยเช่นกัน

Best Practice

- โดยส่วนมากการพัฒนาเว็บแอพมักจะเกี่ยวข้องกับข้อมูลที่ไม่เป็นระเบียบ ซึ่งข้อมูลเหล่านี้อาจทำให้เกิดการผิดพลาดในการใช้งานได้
- ดังนั้นกระบวนการทดสอบระบบจึงมีความสำคัญเป็นอย่างยิ่งเมื่อเทียบกับแอพแบบอื่น
- การทดสอบสำหรับเว็บแอพมักจะประกอบไปด้วย Security, Performance (Load, Stress), Accessibility, Usability, Quality Assurance, และ HTML/CSS Validation
- ส่วนเครื่องมือที่มักจะใช้ในการพัฒนาได้แก่ Bootstrap, JQuery, Ruby on Rails, NodeJS
- ในบางโอกาสเว็บแอพมักจะถูกเปรียบเทียบกับ Native App ที่สร้างมาสำหรับอุปกรณ์นั้นโดยเฉพาะ เว็บแอพจึงเสียเปรียบในบางฟังก์ชันเช่น GPS, Gyroscope, Camera

Performance Testing

- Performance Testing คือการทดสอบเว็บแอพโดยการลองสถานการณ์การใช้งานผ่านอุปกรณ์หลายชนิด การทดสอบสามารถแบ่งออกเป็นหมวด
  - Quantitative คือการวัดเชิงปริมาณเช่น Response Time
  - Qualitative คือการวัดเชิงคุณภาพเช่น ความยืดหยุ่น เสถียรภาพ
- เมื่อพูดถึง Performance คนส่วนมากมักจะคิดถึงความเร็วเป็นหลัก ถึงแม้ว่าความเร็วจะเป็นเรื่องสำคัญ เราควรจะมองภาพใหญ่ของการทดสอบ ซึ่งมีมากกว่าการคลิกแต่ละ Link ในหน้าเว็บเพื่อดูว่าไปยังหน้าที่ถูกต้องหรือไม่

Benefit of Performance Testing

- Performace Testing ช่วยระบุปัญหาที่เกิดขึ้นกับเว็บแอพ การแก้ไขปัญหาเหล่านั้นจะช่วยให้ผู้ใช้งานมีประสบการณ์ที่ดีขึ้น (User Experience)
- หนึ่งในปัญหาที่มักจะได้รับความสนใจคือ ปัญหาคอขวด (Bottleneck) หรือการหยุดชะงักของข้อมูลเนื่องจากความจุที่จำกัด
- ปัญหาคอขวดมักจะเกิดจากการเพิ่มการไหลของข้อมูลโดยฉับพลันและเซิฟเวอร์ไม่สามารถรองรับปริมาณข้อมูล ณ ช่วงเวลานั้นได้
- ความสามารถในการรองรับการขยายตัว (Scalability) จึงเป็นสิ่งจำเป็นที่ต้องทดสอบ มิเช่นนั้นปัญหานี้จะก่อให้เกิดปัญหาอื่นตามมาเช่น Delays, Errors, Memory Leaks

Type of Performance Tests

- Load Test ดูการเปลี่ยนแปลงของ Response Time จาก Workload ที่เพิ่มขึ้น เช่นการเพิ่มจำนวนผู้ใช้งานพร้อมกัน เป้าหมายของการทดสอบนี้คือดูว่าเว็บแอพมีพฤติกรรมอย่างไรในเวลาปกติ (คาดเดาว่าจะมีผู้ใช้งานพร้อมกันกี่คน)
- Stress Test ทดสอบแบบเดียวกันกับ Load Test แต่ภายใต้สถานการณ์ที่ไม่ปกติ เป้าหมายของการทดสอบนี้คือดูว่าเว็บแอพสามารถรองรับการใช้งานจากผู้ใช้พร้อมกันได้มากเท่าไหร่ก่อนที่ระบบจะล่ม
- Load Test และ Stress Test มักจะนำมาใช้ร่วมกันเพื่อมองหาวิธีจัดการกับ Bottleneck ซึ่งจะช่วยเพิ่มจำนวนผู้ใช้งานพร้อมกันได้ด้วย

More Tests

- Endurance Test ทดสอบการทำงานของเว็บแอพในระยะเวลานาน เป้าหมายหลักคือการตรวจสอบ Memory Leak
  - ตัวอย่าง Memory Leak เช่นเราไป Subscribe ฟังก์ชัน SnapshotChanges ของ Document ใน Firestore แต่ลืม Unsubscribe ฟังก์ชันนั้นเมื่อมีการลบ Document จาก Component อื่น
- Spike Test คือ Stress Test ที่มีการเพิ่มและลด Workload สลับไปมาอย่างรวดเร็ว
- Scalability Test ตรวจสอบพฤติกรรมการทำงานของเว็บแอพเมื่อ Workload ค่อย ๆ เพิ่มขึ้น การใช้งานทรัพยากรของเว็บแอพอาจจะแกว่งแม้จำนวน Workload เท่าเดิม
- Volume Test ทดสอบการทำงานของเว็บแอพกับข้อมูลจำนวนมหาศาล

Performance Metrics

- หน่วยในการวัดความสามารถของเว็บแอพก็เป็นสิ่งที่ต้องคำนึงถึง หน่วยเหล่านี้สามารถนำมาใช้เปรียบเทียบและวิเคราะห์การทำงานได้
- Response Time คือระยะเวลาระหว่างการส่ง Request และการรับ Response
- Average Load Time คือค่าเฉลี่ยของ Response Time ของ Request ที่มีผลต่อ User Experience
- Peak Response Time คือ Response Time ที่นานที่สุด หากค่านี้มีค่ามากกว่า Average Load Time แสดงว่าเว็บแอพเรามีปัญหา
- Wait Time คือเวลาที่ Request อยู่ใน Queue ก่อนที่จะถูกประมวลผล

More Metrics

- Requests per Second จำนวน Request ที่สามารถประมวลผลได้ต่อวินาที
- Memory Utilization ปริมาณหน่วยความจำที่ใช้ต่อการประมวลผลหนึ่ง Request
- CPU Utilization เวลาที่ใช้ในการประมวลผลหนึ่ง Request
- Error Rate จำนวน Error ต่อหนึ่ง Request
- Concurrent Users จำนวนผู้ใช้งานพร้อมกัน
- Throughput คือปริมาณข้อมูล (Kilobytes per second) ที่ Bandwidth ใช้ในขณะที่ทดสอบระบบ เนื่องจาก Thoughput เป็นตัวบ่งชี้สมรรถนะของเว็บแอพ การตั้งเป้าหมาย Throughput จึงเป็นสิ่งที่แรกที่ควรทำ

Performance Testing Process

1. กำหนดสิ่งแวดล้อมของการทดสอบ เช่นผู้ใช้มักจะใช้เว็บแอพเราผ่านอุปกรณ์ใดบ้าง แต่ละอุปกรณ์ในปัจจุบันมี Hardware อะไรบ้าง
2. ตั้งเป้าหมายของการทดสอบ คือการตั้งเงื่อนไขการยอมรับในค่าต่าง ๆ เช่น Thoughput, Response Time และ Memory Utilization ถ้าการทดสอบผ่านเงื่อนไขทั้งหมดก็แสดงว่าเว็บแอพของเราพร้อมที่จะให้บริการ การตั้งเป้าหมายที่ต่ำเกินไปอาจจะส่งผลให้การใช้งานจริงไม่เป็นไปตามที่คาดหวัง ในขณะเดียวการตั้งเป้าหมายที่สูงเกินไปอาจจะส่งผลให้เปลืองค่าใช้จ่ายในการอัพเกรดเซิฟเวอร์ ในบางครั้งเราอาจจะใช้เงื่อนไขของเว็บแอพที่ทำงานใกล้เคียงมาใช้ได้

Performance Testing Process (Cont.)

3. วางแผนและออกแบบการทดสอบ เป็นการจำลองเหตุการณ์จริงที่คาดว่าผู้ใช้จะทำ โดยในแต่ละเหตุการณ์จะต้องระบุด้วยว่าจะเก็บข้อมูลแบบไหนบ้าง
4. ใช้ Performance Testing Tool ในการติดตั้งสิ่งแวดล้อมสำหรับการทดสอบ
5. ทดสอบตามแผนที่วางไว้
6. วิเคราะห์ผลลัพธ์ แก้ไขการทดสอบ ทดสอบอีกครั้งตามความเหมาะสม การ Optimize เว็บแอพตามผลลัพธ์จะทำให้การทดสอบในรอบถัดไปดีขึ้น แต่ถ้าหาก Bottleneck เกิดจากตัว CPU เราควรหยุดทดสอบแล้วพิจารณาการอัพเกรดเซิฟเวอร์

Performance Testing Tools

- KeyCDN Tools ทดสอบได้ด้านต่าง ๆ จากการใช้งาน 14 แห่ง
- Google Lighthouse เป็น Open Source แค่ใส่ URL ของเว็บแอพ มีการให้คำแนะนำ
- Selenium เป็น Open Source Framework
- Loadstorm ทดสอบเว็บแอพผ่าน Cloud Service
- WebLOAD ทดสอบเว็บแอพขนาดองค์กรที่มีผู้ใช้งานร่วมหลักพัน
- และอื่น ๆ อีกมากมาย

Optimization ต้องมาก่อนการอัพเกรดเซิฟเวอร์