

Facebook Scalability

บริษัทดังกล่าวให้บริการอะไร

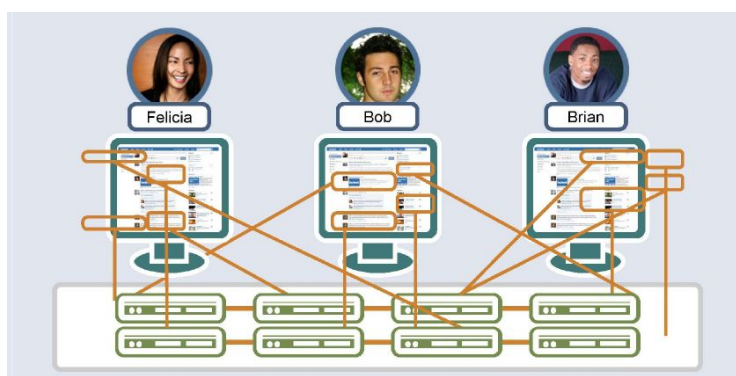
เฟสบุ๊ก(Facebook) เป็นผู้ให้บริการเครือข่ายสังคมออนไลน์ หรือเรียกว่า Social Network โดยจุดประสงค์หลักคือการทำให้ผู้ใช้เข้าถึงกันได้ทั่วโลก โดยมีบริการหลักคือ ฟีดข่าว(News Feed) เป็นบริการที่ทำให้ผู้ใช้ได้มองเห็นข้อมูลของคนอื่นบนเฟสบุ๊ก ที่ได้ทำการโพสต์สถานะ เปลี่ยนแปลงรูปปกโปรไฟล์ อัปโหลดรูป วิดีโอ ลิงค์ ฯลฯ



รูปภาพ ตัวอย่างของบริการฟีดข่าว(News Feed)

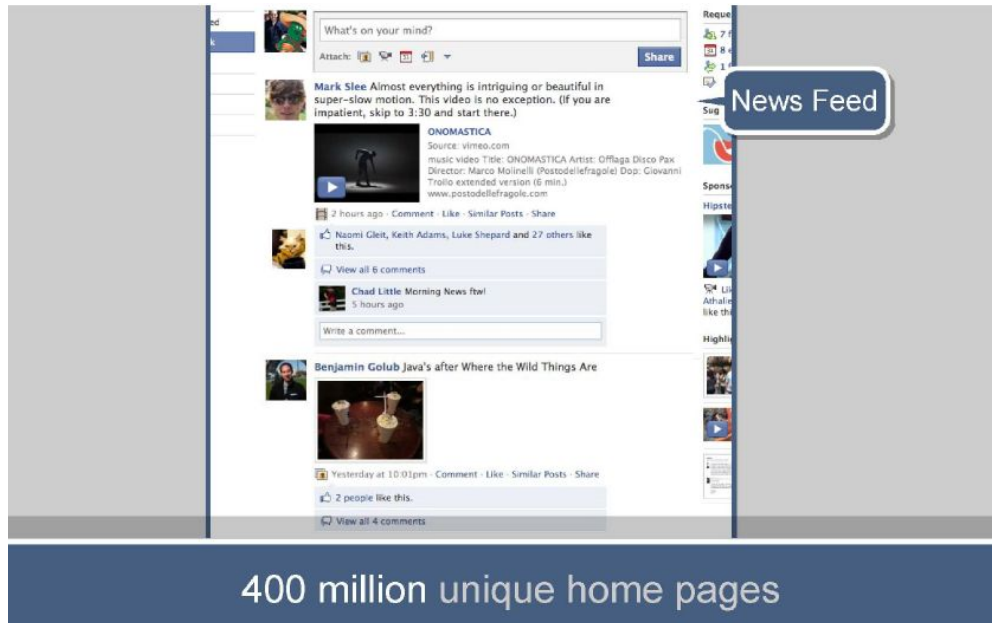
ประเด็นปัญหาเกี่ยวกับ Scalability ที่ประสบในการให้บริการ

- 1) ฟีดข่าวของผู้ใช้ไม่ได้เกิดขึ้นจากข้อมูลที่เก็บด้วยการอ้างอิงจากผู้ใช้เพียงคนเดียว แต่ประกอบด้วยผู้ใช้หลายคนจากหลายฐานข้อมูล ทำให้เกิดความลำบากในการจัดเก็บข้อมูลและเสียเวลาในการดึงข้อมูล



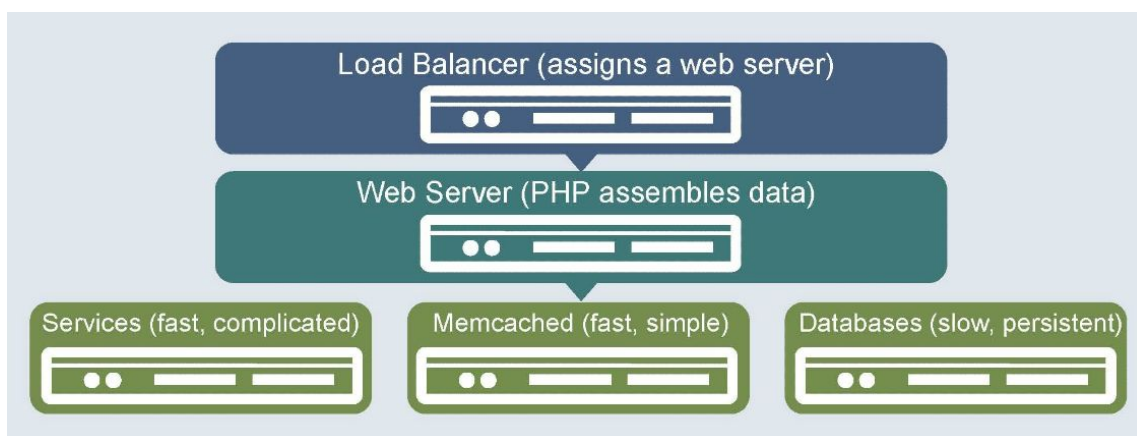
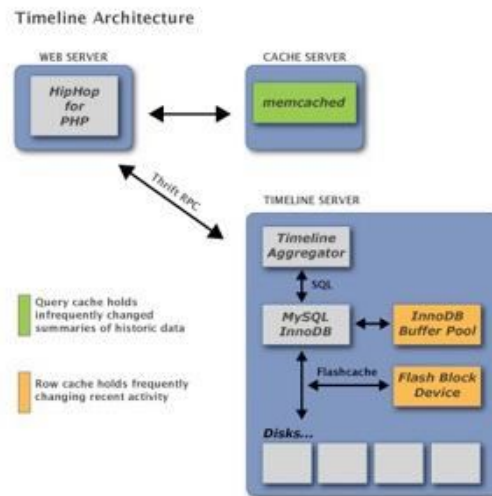
ภาพแสดงลักษณะการเข้าถึงข้อมูลของ ฟีดข่าว

- 2) มีผู้ใช้มากถึง 1 พันล้านคน มีการใช้งานเป็นเวลารวม 8 พันล้านนาที่ต่อวัน มีหน้าเว็บพีดข่าวที่ต่างกัน 400 ล้านหน้าการส่งหน้าเว็บให้แต่ละผู้ใช้อย่างเร็วจึงเป็นไปได้ยาก



- 3) ความละเอียดของรูปภาพและข้อมูลที่เกี่ยวข้องของรูปภาพ ไม่ว่าจะเป็นชื่อภาพ metadata ที่ไม่เกี่ยวข้อง แต่เมื่อรวมกันแล้วจึงทำให้ข้อมูลมีขนาดใหญ่และยากที่จะทำการเก็บหรือตอบสนองให้ผู้ใช้เป็นจำนวนมากและรวดเร็วได้
- 4) ฐานข้อมูลต้อง พร้อมใช้งานตลอดเวลาเนื่องจากมีผู้ใช้ active ตลอดการ migrate duplicate backup ข้อมูลมีโอกาสเสี่ยงทำให้เกิด failure ของฐานข้อมูลได้

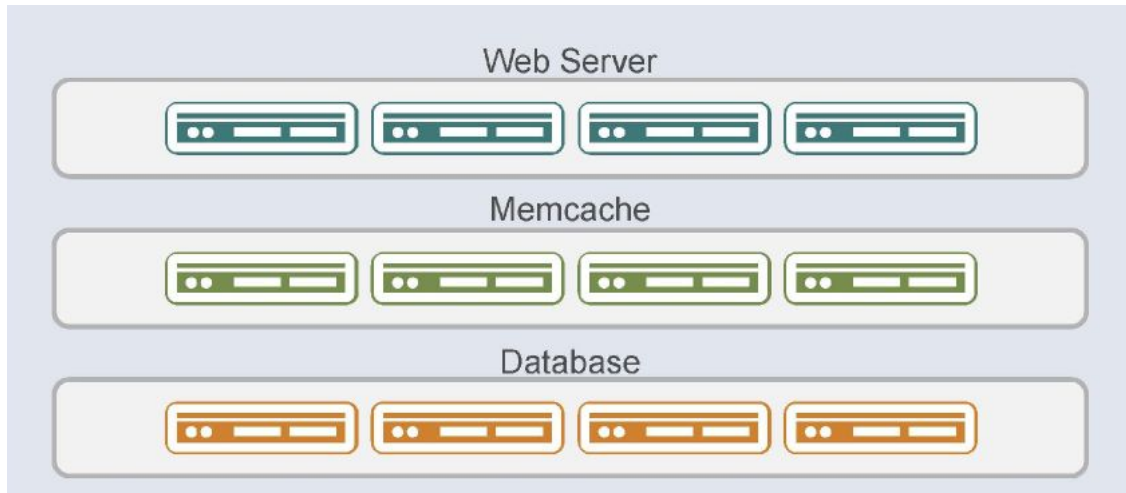
เทคนิควิธี หรือสถาปัตยกรรมที่ใช้ในการแก้ปัญหา



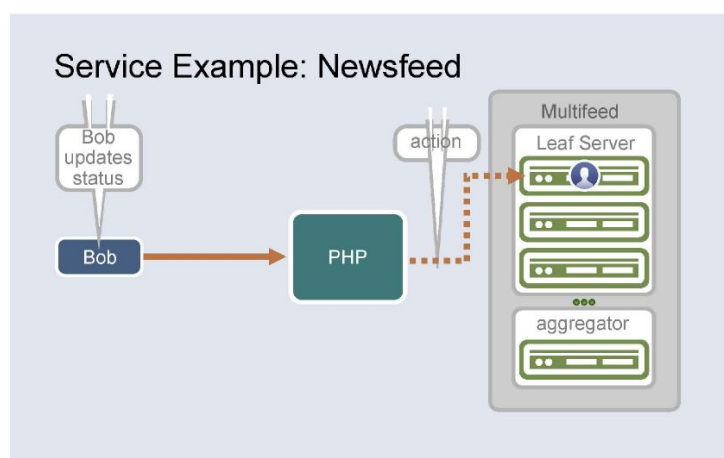
ภาพแสดงโครงสร้างสถาปัตยกรรมอย่างคร่าวๆของเฟสบุ๊ค

- 1) เพิ่มประสิทธิภาพ Web Server ที่ทำจาก PHP ด้วยการทำ framework ขึ้นมาชื่อว่า HipHop เนื่องจากภาษา PHP เป็นภาษาที่เสียเวลาในช่วง runtime ค่อนข้างมาก HipHop จึงช่วยในการแปลงภาษาให้เป็น C++ ซึ่งเป็นภาษาที่เร็วในช่วง runtime มากกว่า ก่อนที่จะ deploy code
- 2) ปรับรูปแบบการเก็บข้อมูลใน mySQL ไม่ทำแบบ Normalization แต่จัดการเก็บข้อมูลให้เหมาะสมกับที่ต้องการเรียกใช้ ไม่มีการเก็บข้อมูลที่จะต้อง join ตารางเก็บข้อมูลแบบ HashTable

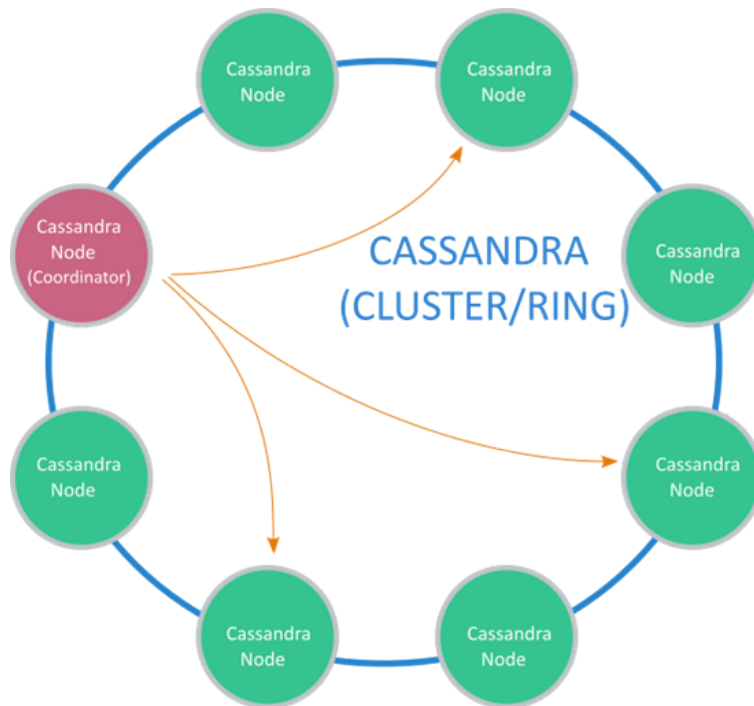
- 3) ทำ Index Cache ด้วย Memcache (PHP Extension) เพื่อลดเวลาการประมวลผลเพื่อหาข้อมูลจากฐานข้อมูลที่ช้าโดยตรง และนำ index ที่ได้ ไปหาใน Database อีกครั้งซึ่งเก็บข้อมูลแบบ Hash Table



- 4) - ตัดรูปแบบความละเอียดของภาพออก เก็บเพียงความละเอียดของภาพ 3 ชุด คือ เล็ก กลาง และใหญ่
- ตัด metadata ที่ไม่เกี่ยวข้องออกทั้งหมด เก็บเพียง data ของรูป
 - กระจายรูปภาพเก็บใน CDN เพื่อลด Network latency
 - ทำ Write Back ในครั้งแรกที่อัปโหลด
- 5) ทำ Horizontal Scaling ในการเก็บ Feed เปรียบเสมือนฟีด 1 คนคือ 1 เครื่อง และมีเครื่องที่เรียกว่า aggregator ทำการรวมข้อมูลของแต่ละคนที่เกี่ยวข้องมาเป็น feed



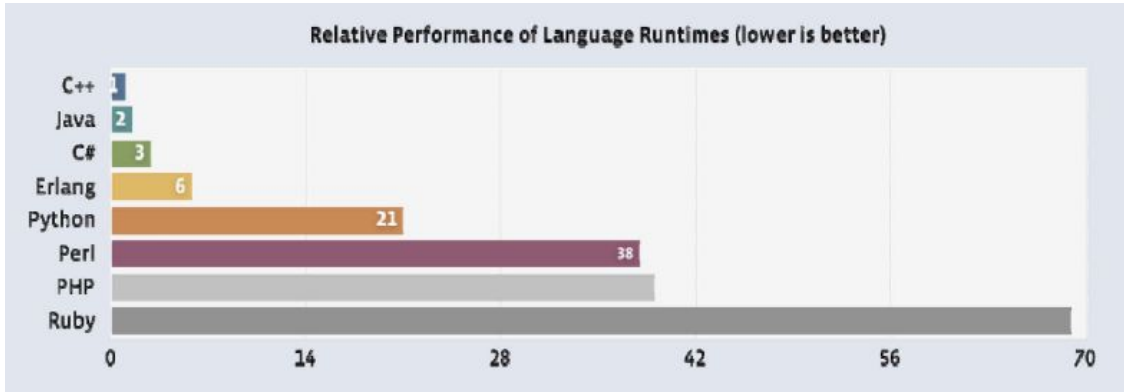
6) ใช้ Apache Cassandra ในการจัดการฐานข้อมูลเพื่อกระจาย Work load, Database, ป้องกันการเสียหายของข้อมูล รวมถึงการเพิ่มเดตาเบสเข้ามาในระบบ



ภาพแสดงการ Distributed Database

ผลที่ได้เมื่อใช้เทคนิควิธีดังกล่าว

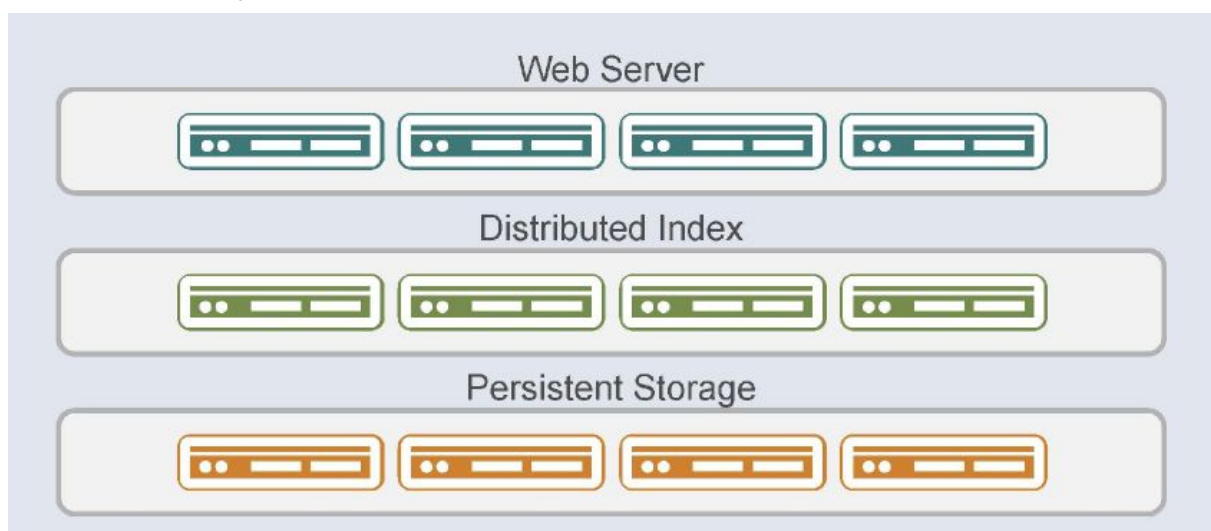
จากข้อ 1) การใช้ HipHop แปลง PHP เป็น C++ ผลที่ได้คือเสียเวลาในช่วง runtime น้อยลงอย่างเห็นได้ชัด



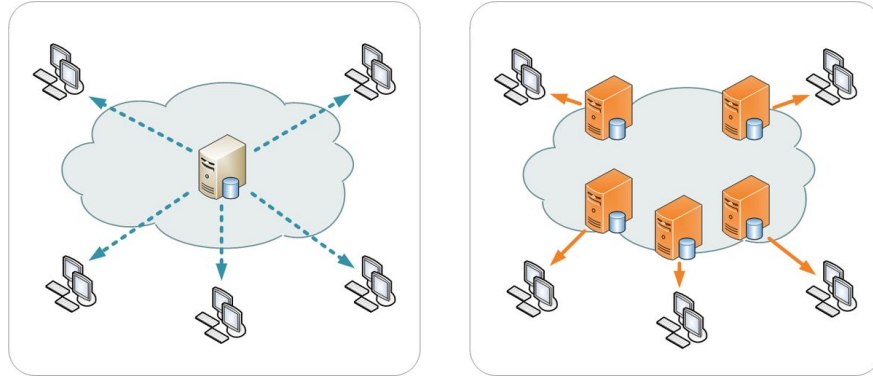
ภาพแสดงการเปรียบเทียบประสิทธิภาพของภาษาในช่วง runtime

จากข้อ 2) การ Denormalize ข้อมูล ผลที่ได้คือ ลดเวลาของ I/O ที่จะต้องใช้ในการ join ตาราง ง่ายต่อการโยกย้ายข้อมูล และง่ายต่อการขยายเครื่องในแบบ Horizontal Scaling

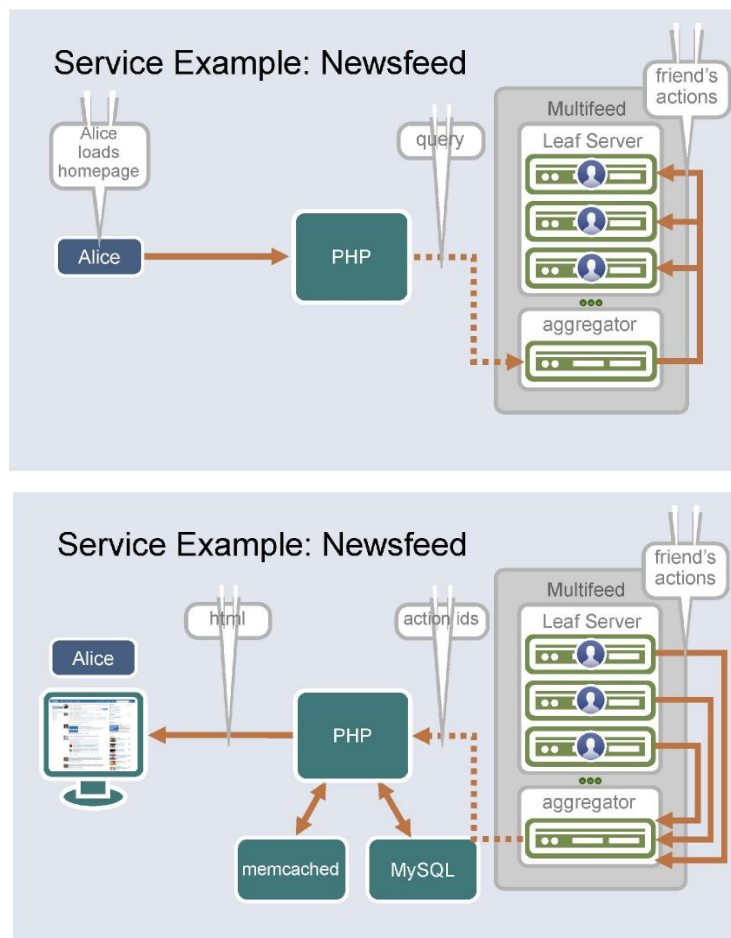
จากข้อ 3) การทำ Memcache ผลที่ได้คือ ลดเวลาในการเทียบหาข้อมูล เนื่องจาก Memcache มีความรวดเร็วในการหา index ของข้อมูลที่ต้องการ และนำไปหาในฐานข้อมูลที่เก็บในรูปแบบของชุดข้อมูลที่เรียกใช้ได้โดยตรง โดยไม่ต้องเชื่อมตาราง เนื่องจากเก็บข้อมูลเป็น Hash Table



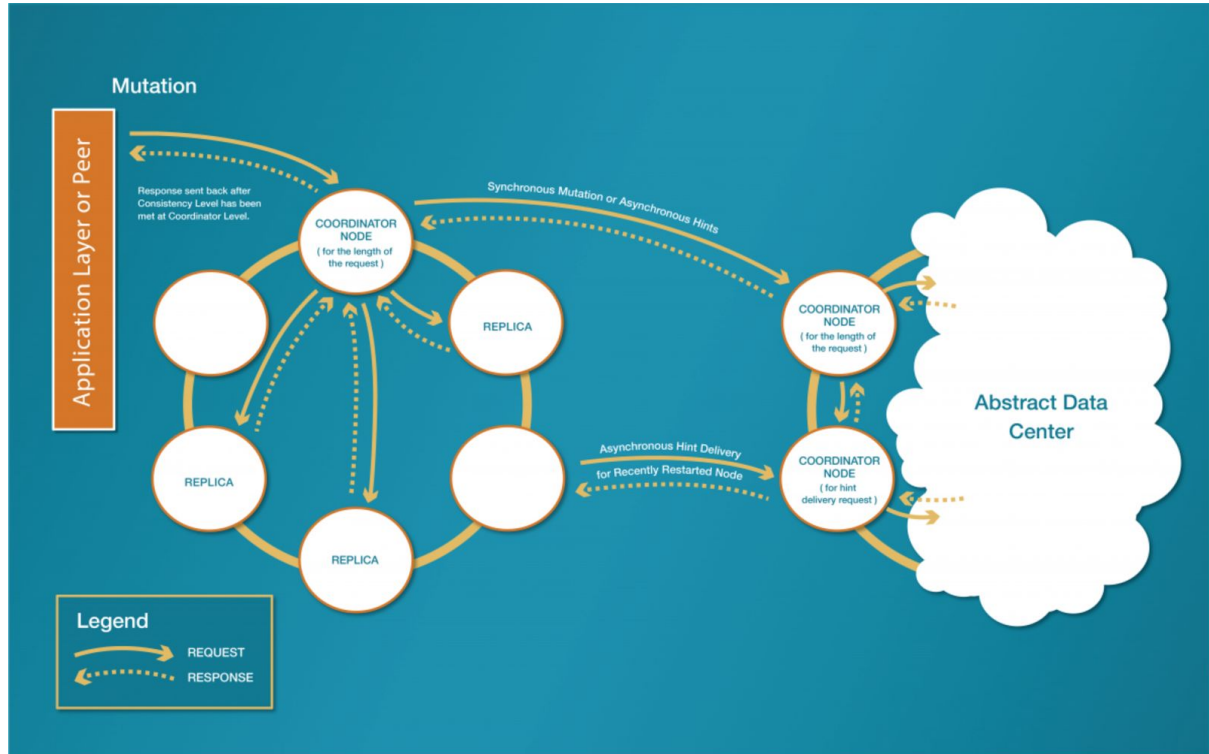
จากข้อ 4) การทำ CDN ผลที่ได้คือ รูปภาพซึ่งเป็นส่วนที่มีข้อมูลขนาดใหญ่ไม่หนัก traffic มาที่เครื่อง Server โดยตรงเนื่องจากมี CDN ช่วยในการเก็บรูปภาพและส่งถึงผู้ใช้ ที่ใกล้เคียงได้เร็วกว่า



จากข้อ 5) การ Distribute Feed แต่ละเครื่อง ผลที่ได้คือข้อมูล Feed สามารถ Scale เครื่องได้ง่ายขึ้น



จากข้อ 6) การใช้ Apache Cassandra Database Scaling ส่งผลให้ฐานข้อมูลไม่จำเป็นต้องมี Master หรือ Slave Workload กระจายไปยังฐานข้อมูลแต่ละก้อน ป้องกันการล้มของฐานข้อมูลจากการใช้ Peer-to-Peer Replication อยู่ตลอดเวลา

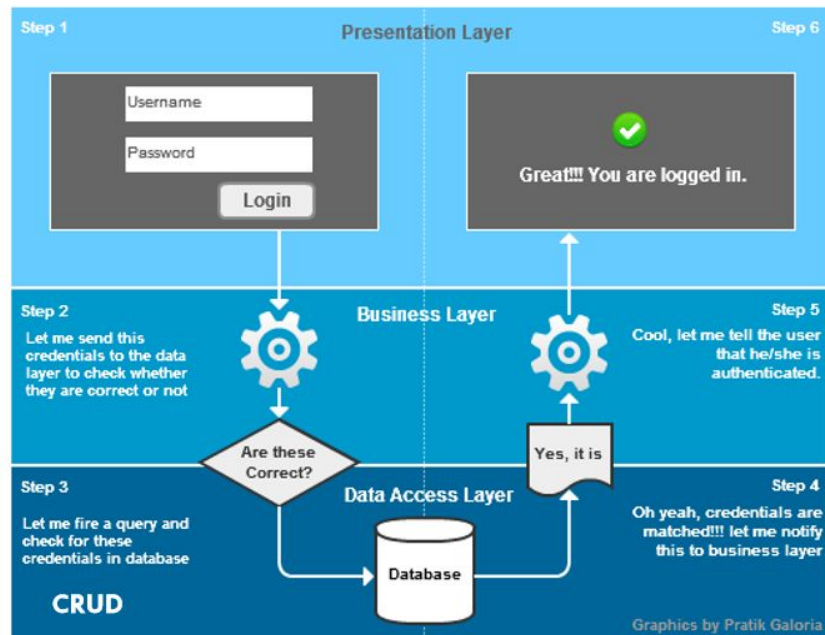


อภิปรายความสัมพันธ์กับเนื้อหาที่เรียนในชั่วโมงบรรยาย

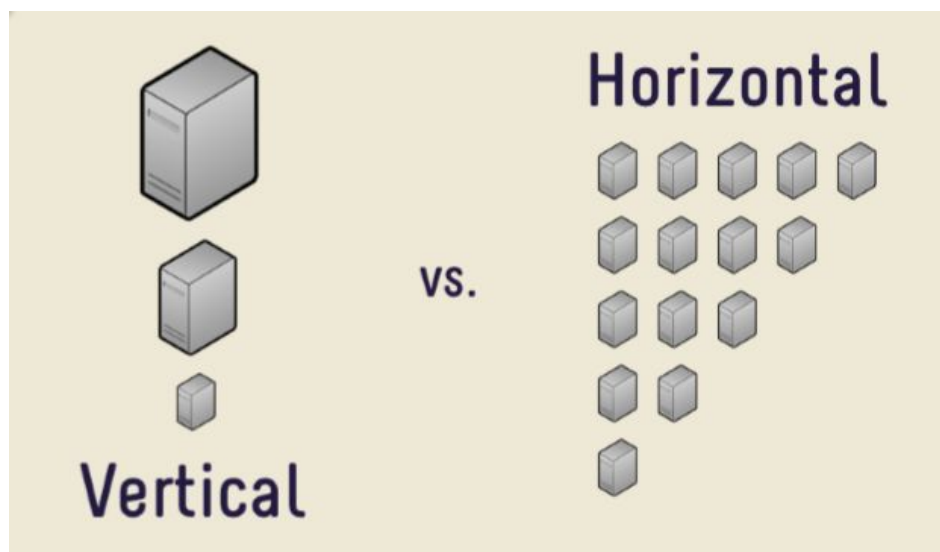
จากการศึกษา Facebook Scalability ทำให้พบว่ามีคุณสมบัติสอดคล้องกับเนื้อหาที่เรียนดังต่อไปนี้

- 1) การแยกชั้น Tier ในการ Scale เป็น 3 ชั้นอย่างคร่าวดังรูปต่อไปนี้

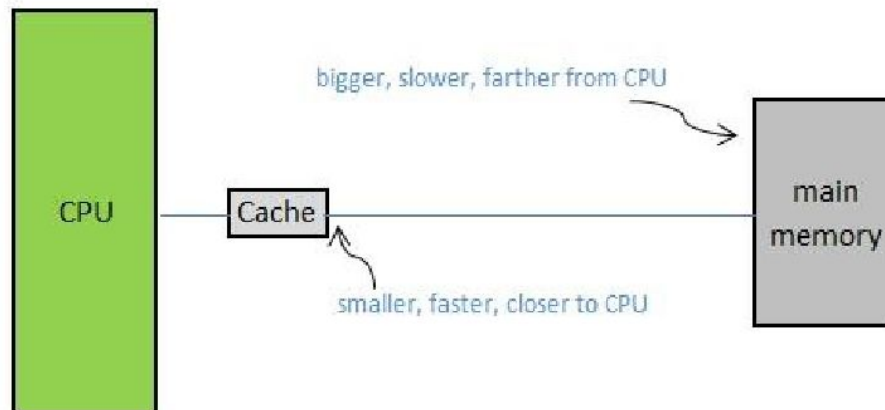
REVISITED THREE-TIER WEB ARCHITECTURE



- 2) มีการ Scale แบบ Horizontal Scaling



3) มีการทำ Caching ซึ่งเป็น Hardware ที่รวดเร็วกว่าการเก็บลง Storage โดยตรง



4) ถึง Facebook จะใช้ mySQL แต่การเก็บข้อมูลก็เป็นลักษณะของ lack of relational และทำให้กลายเป็น HashTable เหมือนกับเรื่องที่เรียน non-SQL



Relational data model

Highly-structured table organization with rigidly-defined data formats and record structure.

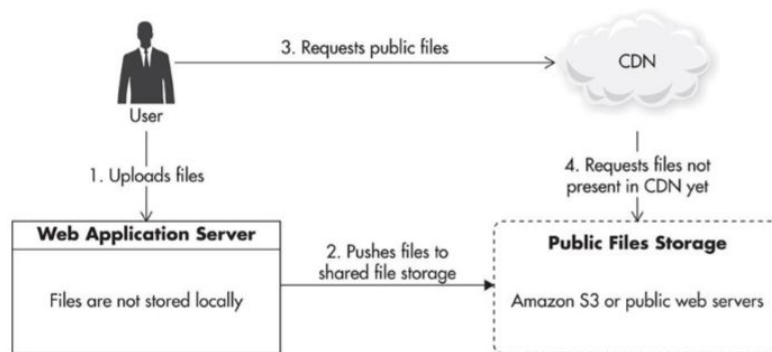


Document data model

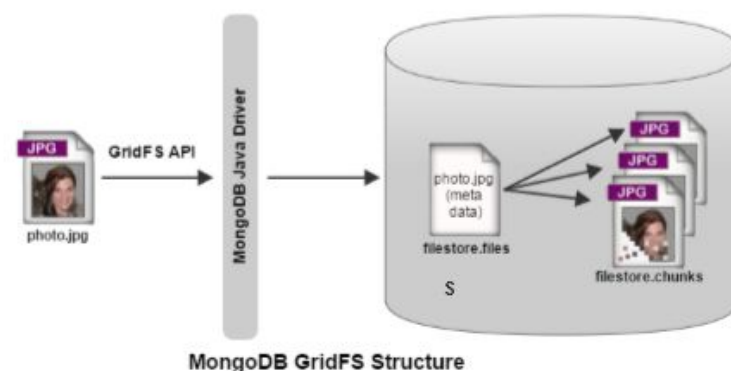
Collection of complex documents with arbitrary, nested data formats and varying "record" format.

5) มีการใช้ CDN (Cloud Delivery Network) ในการ Cache Static File อย่างรูปภาพ

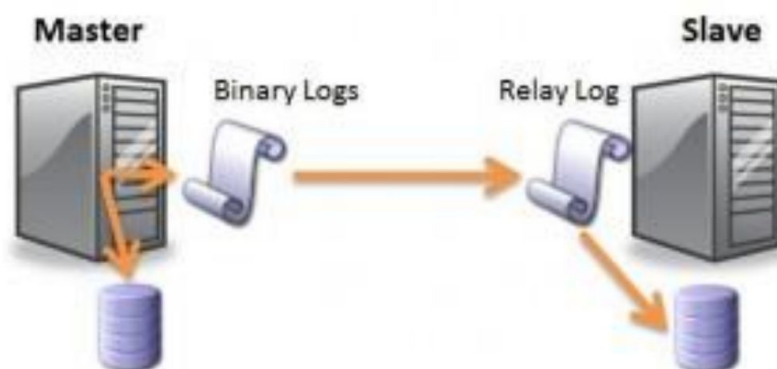
DISTRIBUTED STORAGE AND DELIVERY OF PUBLIC FILES



6) มีการ Manage File ก่อนทำการเก็บข้อมูล เช่นรูปภาพที่ Facebook ตัด metadata ที่ไม่จำเป็นออกไป และทำการเก็บเพียง 3 resolutions



7) มีการทำ SQL Scaling เพื่อป้องกันการเสียหายของฐานข้อมูล และรองรับการขยายขนาดของฐานข้อมูล



แหล่งอ้างอิงที่น่าเชื่อถือ

- 1) <http://highscalability.com/blog/2012/1/23/facebook-timeline-brought-to-you-by-the-power-of-denormaliza.html>
- 2) https://www.facebook.com/note.php?note_id=10150468255628920
- 3) https://link.springer.com/chapter/10.1007/978-3-319-16486-1_75
- 4) <https://www.infoq.com/presentations/Scale-at-Facebook>
- 5) <https://techcrunch.com/2017/06/27/facebook-2-billion-users/>
- 6) <https://support.hostatom.com/knowledgebase.php?action=displayarticle&id=78>
- 7) <https://www.facebook.com/notes/facebook-engineering/cassandra-a-structured-storage-system-on-a-p2p-network/24413138919/>