



## **Flickr Scalability**

โดย

นายธนกร สว่างโลก

โครงการในรายงานวิชาคพ.447 นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

วิทยาศาสตร์บัณฑิตสาขาวิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์

ปีการศึกษา 2560

## สารบัญ

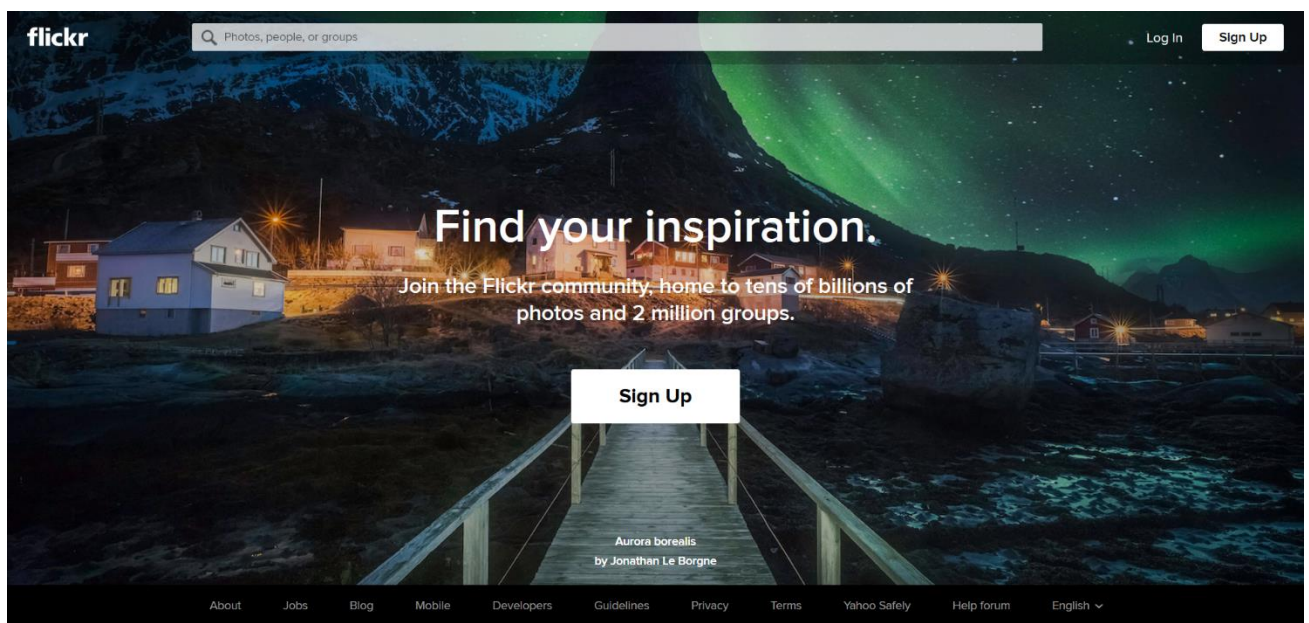
สารบัญ.....	1
สารบัญรูป .....	2
1.Flickr ให้บริการอะไร? .....	3
2.ประเด็นปัญหาเกี่ยวกับ Scalability ที่ประสบในการให้บริการ .....	4
1. การอัปโหลดรูปภาพมากกว่า 4 แสนภาพต่อวัน(Write problem) .....	4
2. การเข้าใช้งานเว็บไซต์มากกว่า 4 พันล้าน queries ต่อวัน(Read Problem).4	
3. การค้นหารูปภาพ(Search problem).....	4
3.เทคนิควิธี หรือสถาปัตยกรรมที่ใช้ในการแก้ปัญหา .....	5
1. Shard .....	5
2. Master-Master topology.....	6
3. Cache & server.....	6
4. Load balancer .....	6
5. Big search engine.....	6
6. Dual tree central Database.....	6
4.ผลที่ได้จากการใช้เทคนิควิธีแก้ปัญหา .....	7
5.ความสัมพันธ์กับเนื้อหาที่เรียนในชม.บรรยาย .....	8
เอกสารอ้างอิง .....	11

## สารบัญรูป

รูปภาพ 1 หน้าหลักของ Flickr .....	3
รูปภาพ 2 สถาปัตยกรรมของ Flickr .....	5
รูปภาพ 3 Horizontal Scaling .....	8
รูปภาพ 4 การทำงานแบบ stateless .....	8
รูปภาพ 5 master-master ring topology .....	9
รูปภาพ 6 shard.....	9
รูปภาพ 7 cache.....	10
รูปภาพ 8 Reverse-proxy.....	10

## 1.Flickr ให้บริการอะไร?

Flickr อ่านว่า “ฟลิคเกอร์” เป็นชื่อของเว็บไซต์ ให้บริการการฝากรูปภาพ วีดิโอไว้ผ่านระบบดิจิทัล ได้รับความนิยมมาก มีผู้ใช้งานเว็บไซต์มากกว่า 4 พันล้าน queries ต่อ 1 วัน หน้าหลักของ Flickr แสดงในรูปภาพที่ 1 โดยผู้ใช้งานสามารถแบ่งปันข้อมูลรูปภาพ วีดิโอกับบุคคลอื่นได้ทั่วโลก สิ่งที่ทำให้ ฟลิคเกอร์แตกต่างจากเว็บไซต์อัปโหลดรูป วีดิโอทั่วไปคือการมี tag ไว้เชื่อมโยงรูปภาพที่คล้ายกัน ประเภทเดียวกัน เพื่อให้สามารถค้นหาได้โดยง่าย มีระบบ Organizer คอยจัดกลุ่มภาพที่ความคล้ายกันด้วย tag , Date และบริการอื่นอีกมากมาย



รูปภาพ 1 หน้าหลักของ Flickr

## 2.ประเด็นปัญหาเกี่ยวกับ Scalability ที่ประสบในการให้บริการ

Flickr เนื่องจากได้รับความนิยมสูงมาก มีผู้ใช้งานเว็บไซต์มากกว่า 4 พันล้าน queries ต่อ 1 วัน มีรูปภาพทั้งหมดมากกว่า 2 พันล้านบนเว็บไซต์ นอกจากนี้ รูปภาพมากกว่า 400,000(สี่แสน) ภาพ ถูกเพิ่มเข้ามาในระบบต่อ 1 วัน และบริการเหล่านี้ยังเพิ่มขึ้นเรื่อย ๆ ทำให้เจอปัญหาเกี่ยวกับ scalability ดังต่อไปนี้

### 1. การอัปโหลดรูปภาพมากกว่า 4 แสนภาพต่อวัน(Write problem)

เนื่องจากมี active user หลักร้อยล้านหรือพันล้าน ทำการอัปโหลดรูปภาพขึ้นไปทีเว็บไซต์ 4 แสนภาพต่อวัน รูปแบบการเก็บข้อมูลต้องรวดเร็ว รองรับการเขียนข้อมูลขนาดยักษ์ได้

### 2. การใช้งานเว็บไซต์มากกว่า 4 พันล้าน queries ต่อวัน(Read Problem)

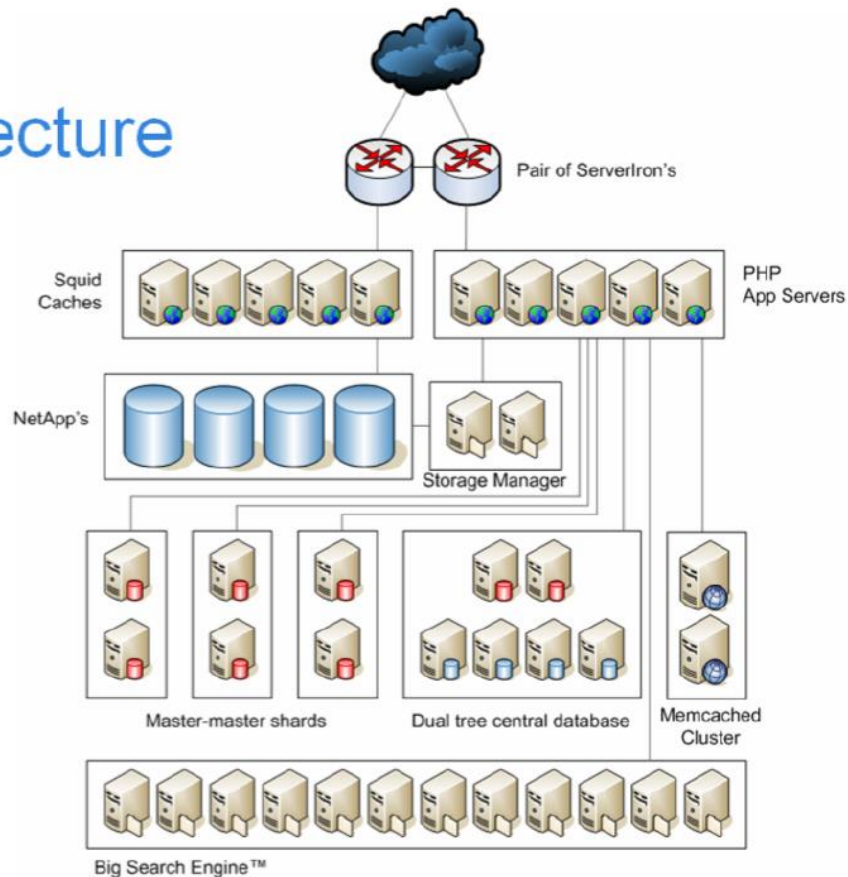
จำนวนคนใช้งานสูงมาก คิดเป็น 1 วินาที มีการquery เข้ามาถึง 46296 request(จากผู้ใช้ทั้งหมดใน 1 วัน) เลยทีเดียว เว็บไซต์จำเป็นต้องรองรับการใช้งานของผู้ใช้โดยไม่ล่าช้าไปก่อน

### 3. การค้นหารูปภาพ(Search problem)

เนื่องจาก Flickr มี Tag ไว้จัดเชื่อมโยงรูปภาพที่คล้ายกัน เหมือนกัน หรือกลุ่มเดียวกันไว้ด้วยกัน การค้นหารูปภาพ ต้องค้นหาจากรูปภาพมหาศาลและต้องค้นหามาแสดงผลผู้ใช้ได้รวดเร็วด้วย

### 3.เทคนิควิธี หรือสถาปัตยกรรมที่ใช้ในการแก้ปัญหา

## Flickr Architecture



รูปภาพ 2 สถาปัตยกรรมของ Flickr

#### 1. Shard

เพื่อรองรับการเขียน อ่าน ค้นหา ไฟล์รูปภาพให้รวดเร็ว รองรับการทำงานขนาดยักษ์ได้ ข้อมูลในฐานข้อมูลหลักถูกแบ่ง แต่ละ shard ทำ Master-Master Ring Replication เพื่อให้เขียนข้อมูลได้ดีขึ้นมากขึ้น assign user ให้แต่ละ shard ด้วย random number account id แต่ละ shard จะมี user 400K+ เท่านั้น กำหนดการทำงานของแต่ละ shard ไว้ไม่ให้เกิน 50% อาจจะมีเกินในกรณีที่ maintenances หรือ shutdown ถ้าเกินจะทำการ เพิ่ม หรือ upgradeจากการแบ่งข้อมูลเป็น shard ทำให้สามารถรับมือการอ่านเขียนขนาดยักษ์ได้

## 2. Master-Master topology

สถาปัตยกรรมก่อนหน้านี้ของ flickr ใช้รูปแบบ Master slave topology ซึ่งแน่นอนว่า การเข้าใช้มากกว่า 4 พันล้านต่อวัน ทำให้ระบบล่มไปบ่อย และมีการอัปโหลดรูปภาพ (write) มากกว่า 4 แสนรูป เพื่อให้การอัปโหลดสามารถทำได้เร็วมากขึ้น จึงเพิ่มตัวเขียนข้อมูลลงไป ทำให้Flickr เปลี่ยนมาใช้ Master-Master topology แต่ละ shard แยกข้อมูลออกจากกันชัดเจน การอ่าน เขียนของ user 1 คนทำใน shard เดียว แก้ปัญหาการ comment ลงคนละ shard ด้วย distributed transaction ถ้าข้อมูลผิดพลาดจะ roll back กลับไป

## 3. Cache & server

เพื่อลดภาระของ server ลงให้เยอะที่สุด มีการทำ cache เกิดขึ้น ทำ Squid (reverse-proxy) กับไฟล์ html หรือ รูปภาพ เพื่อลดภาระการประมวลผลของserver และ memcache เพื่อประหยัดเวลาการ query ข้อมูลใน database ไม่ต้อง query ซ้ำกันหลายครั้ง ให้แสดงผลได้เร็วขึ้น สำหรับข้อมูลที่เป็น static ใช้ dedicate server

## 4. Load balancer

ใช้ pair of Serveriron เป็น load balancer กระจายงานไปให้ server ทำให้แสดงผลให้ผู้ใช้งานได้เร็วขึ้น

## 5. Big search engine

การ search tag , organize, batch tag change จะใช้ shard search สำหรับการค้นหาแบบอื่น จะใช้ yahoo search เพื่อให้ค้นหาข้อมูลได้รวดเร็วยิ่งขึ้น

## 6. Dual tree central Database

เป็นฐานข้อมูลที่ไว้เก็บข้อมูลของ user ว่าข้อมูลของ user อยู่ใน shard ไหน สถาปัตยกรรมแบบ ช่วยให้แสดงผลได้เร็วขึ้น เพราะไม่ต้องไปเสียเวลาค้นหาว่าผู้ใช้คนนี้อยู่ใน shard ไหนทุกครั้ง

#### 4.ผลที่ได้จากการใช้เทคนิควิธีแก้ปัญหา

1. Flickr สามารถรองรับการเข้าใช้งานขนาดยักษ์ได้มากขึ้น
2. Flickr สามารถแสดงผลการค้นหารูปได้เร็วมากขึ้น
3. Flickr สามารถอ่านเขียนข้อมูลไฟล์ได้เร็วมากขึ้น
4. ฐานข้อมูลเข้าถึงได้ตลอดเวลาและรวดเร็วยิ่งขึ้น เพราะ แบ่งเป็น shardไว้
5. เว็บไซต์ไม่ล่มบ่อย ใช้งานได้นานขึ้น
6. สามารถทำ scaling เพิ่มได้ในอนาคต



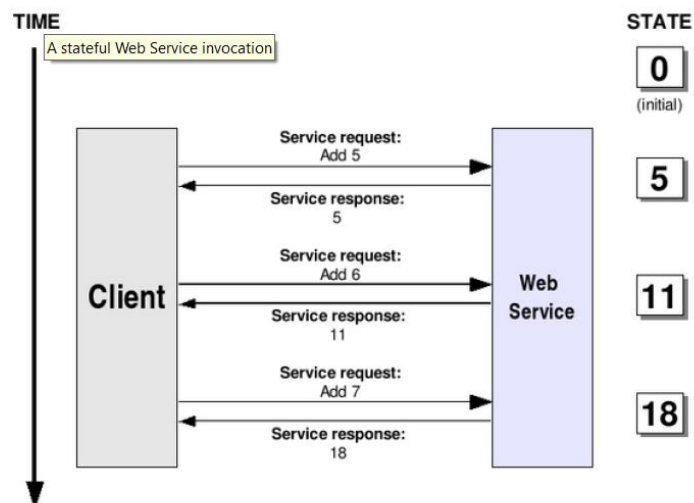
## 5.ความสัมพันธ์กับเนื้อหาที่เรียนในชม.บรรยาย

1. มีการทำ horizontal scaling เพื่อขยายการทำงานให้รองรับผู้ใช้ได้มากขึ้น



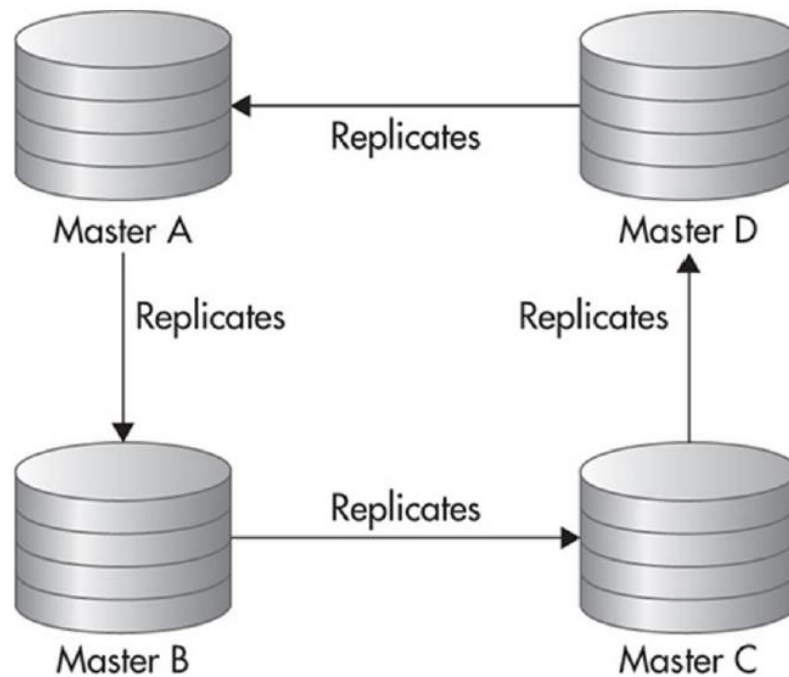
รูปภาพ 3 Horizontal Scaling

2. Server มีการทำงานแบบ stateless



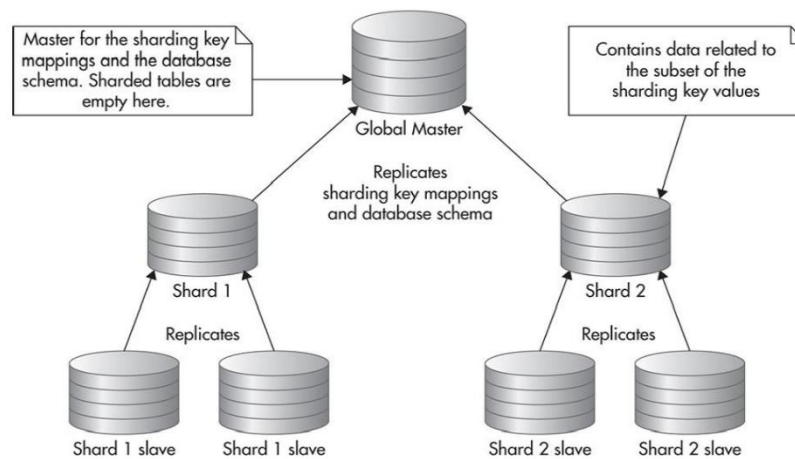
รูปภาพ 4 การทำงานแบบ stateless

3. มีการใช้ load balancer เพื่อกระจายการทำงานของ server ไม่ให้หนักเกินไป
4. มีการใช้ master-master ring topology เพื่อให้เขียนข้อมูลได้มากขึ้น



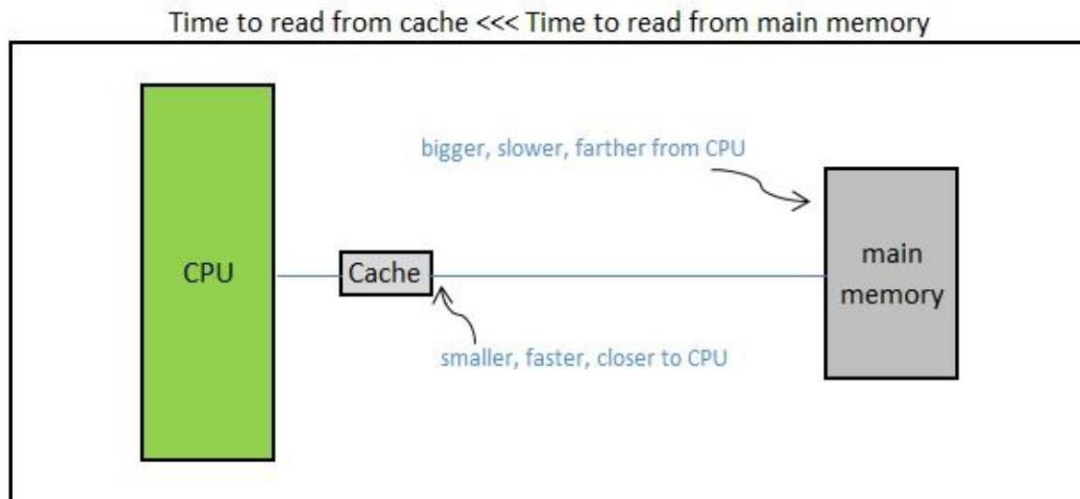
รูปภาพ 5 master-master ring topology

5. มีการทำ shard เพื่อแบ่งฐานข้อมูลให้ทำกระบวนการอ่านเขียน ค้นหาได้รวดเร็วยิ่งขึ้น



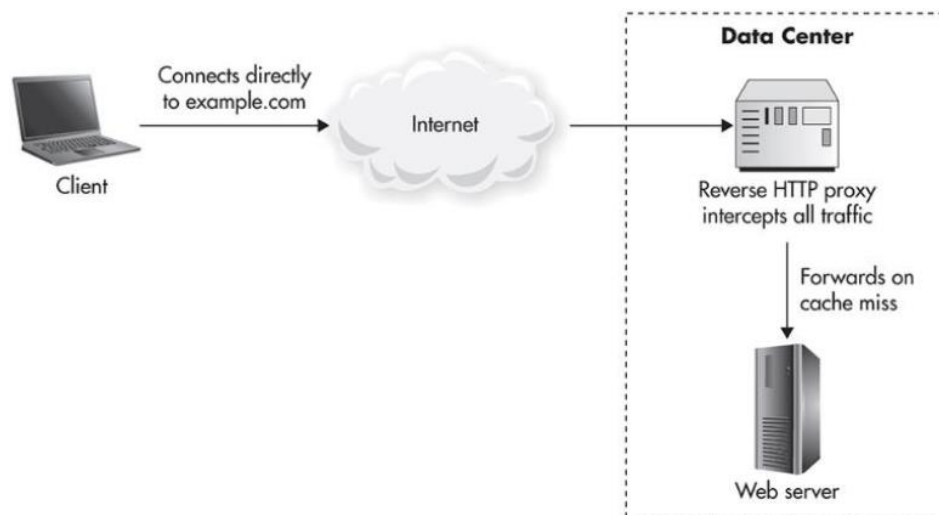
รูปภาพ 6 shard

6. มีการทำ cache เพื่อให้ค้นหาข้อมูลได้เร็วยิ่งขึ้น



รูปภาพ 7 cache

7. มีการทำ reverse-proxy เพื่อลดภาระการทำงานของฝั่ง server



รูปภาพ 8 Reverse-proxy

## เอกสารอ้างอิง

Modify. *Flickr คืออะไร ไว้ใช้ทำอะไร*[ออนไลน์]. 2013. สืบค้นจาก:

<https://www.modify.in.th/3851>

Jose Vieitez. *What is the software architecture of Flickr?* [ออนไลน์]. 2016. สืบค้นจาก:

<https://www.quora.com/What-is-the-software-architecture-of-Flickr>

Josh Lowensohn. *Flickr hits 2 billion shots*

*Flickr's got a lot of pictures, but what's the big picture when it comes to photo uploading?*[ออนไลน์]. 2007. สืบค้นจาก: <https://www.cnet.com/news/flickr-hits-2-billion-shots/>

highscalability. *Flickr Architecture*[ออนไลน์]. 2007. สืบค้นจาก:

<http://highscalability.com/blog/2007/11/13/flickr-architecture.html>

Dathan Vance Pattishall. *Federation at Flickr: Doing Billions of Queries Per Day*

[ออนไลน์]. สืบค้นจาก: <https://www.scribd.com/doc/2592098/DVPmysqlucFederation-at-Flickr-Doing-Billions-of-Queries-Per-Day> [เข้าถึงเมื่อ Dec 19,2017]

Mikhail Panchenko. *The Evolution of the Flickr Architecture*[ออนไลน์].2010. สืบค้น

จาก: <https://www.infoq.com/presentations/Flickr-Architecture>

Apiwat Tatsanakitti. หลักการทำงานของ Proxy Server และ Reverse Proxy [ฉบับ  
ละเอียดอ่อน][ออนไลน์].2016. สืบค้นจาก:

<http://network99public.blogspot.com/2016/06/proxy-server-reverse-proxy.html>