

ฐานข้อมูล NewSQL และแนวโน้มการใช้งาน NewSQL Databases and Usability Trends

ภูมินทร์ บุญเฮ้า*
PUMIN BOONHAO*

บทคัดย่อ

ในปัจจุบันเทคโนโลยีของฐานข้อมูลมีการพัฒนาก้าวหน้าต่อเนื่องไปอย่างรวดเร็ว ปัจจัยที่ทำให้เทคโนโลยีของฐานข้อมูลต้องมีการพัฒนาอยู่ตลอดคือ ความต้องการใช้งานฐานข้อมูลเพื่อให้รองรับการทำงานของแอปพลิเคชันที่หลากหลาย ประเภทของข้อมูลที่หลากหลาย การประมวลผล และการวิเคราะห์ข้อมูลขนาดใหญ่ที่รวดเร็วมากขึ้น รวมถึงการขยายฐานข้อมูลในอนาคตหลังจากขึ้นใช้งานแอปพลิเคชัน

ความต้องการฐานข้อมูลที่มีขนาดใหญ่เพิ่มมากขึ้นแต่ฐานข้อมูล RDBMS (SQL) หรือฐานข้อมูล NoSQL ยังไม่สามารถตอบโจทย์ความต้องการในปัจจุบันได้ ทำให้ต้องพัฒนาฐานข้อมูลที่เรียกว่าฐานข้อมูล NewSQL ขึ้นมาเพื่อตอบสนองความต้องการในการใช้งาน การรวมคุณสมบัติที่โดดเด่นของฐานข้อมูล RDBMS (SQL) และฐานข้อมูล NoSQL เข้าด้วยกัน เช่น คุณสมบัติ Relational Databases, Distributed Databases, การวิเคราะห์ข้อมูลขนาดใหญ่ (OLAP/OLTP), ACID แต่อย่างไรก็ตามฐานข้อมูล NewSQL ก็ยังแบ่งประเภทออกเป็น Historical, Real-Time, Streaming, และ Timestamp เพื่อการเลือกใช้งานให้เหมาะสมมากที่สุด

หัวใจสำคัญของการเลือกใช้งานฐานข้อมูลคือ ประเภทของข้อมูลที่ใช้ในการออกแบบและการเลือกใช้งานฐานข้อมูลให้เหมาะสม รวมไปถึงความคุ้มค่าที่จะลงทุน เพราะไม่มีเทคโนโลยีใดที่ไม่มีข้อเสีย จุดอ่อนของ ฐานข้อมูล NewSQL คือข้อมูลกรณีศึกษา (Use Case) และชุมชนของผู้ใช้งาน (Community) มีน้อย ทำให้ผู้ที่ใช้งานฐานข้อมูล NewSQL ต้องมีความรู้และความเข้าใจในเรื่องของฐานข้อมูลเป็นอย่างดี ดังนั้นองค์กรขนาดใหญ่จะเปลี่ยนรากฐานองค์กรย่อมต้องมีความมั่นใจว่าเลือกใช้เทคโนโลยีได้ถูกต้องและเหมาะสม

คำสำคัญ: ฐานข้อมูล RDBMS (SQL), ฐานข้อมูล NoSQL, ฐานข้อมูล NewSQL

* ฝ่ายสารสนเทศ คณะแพทยศาสตร์ศิริราชพยาบาล มหาวิทยาลัยมหิดล

* Siriraj Information Technology Department, Faculty of Medicine Siriraj Hospital, Mahidol University

* E-mail: pumin.boo@mahidol.ac.th

Abstract

Currently, the database technology has been developing rapidly. The ubiquitous database used to support the operation of various applications processing of big data analytics and scale up database in the future. Widespread usage has resulted in the development of database technology at all times.

The demand for using scale-up databases is increasing in the RDBMS (SQL) Databases and NoSQL Databases. Resulting in the development of NewSQL Databases is combining the distinctive features of RDBMS (SQL) Databases and NoSQL Databases, such as relational databases, distributed databases, big data analytics (OLAP/OLTP) and ACID. Conversely, NewSQL Databases classified into historical, real-time, streaming, and timestamp in order to select the most appropriate usage.

The major key to choosing database is the type of information in design including the selection of appropriate databases and the worth for investment because all of the technologies have disadvantages. The weakness of NewSQL Databases is lack of use cases and the community to exchange new knowledge. This reason NewSQL Databases users must have outstanding knowledge and understanding. Therefore, large organizations will be change to new technology, must select the appropriate technology in the current organizational context.

Key word: RDBMS (SQL) Databases, NoSQL Databases, NewSQL Databases

บทนำ

การดำเนินชีวิตประจำวันในปัจจุบันนั้น จะมีเทคโนโลยีสารสนเทศมาเกี่ยวข้องในทุกๆด้านอย่างแนบแน่น และเทคโนโลยีสารสนเทศก็ได้เข้าไปเป็นส่วนหนึ่งในการขับเคลื่อนองค์กรต่างๆ จนถึงระดับประเทศ จะเห็นได้จากการประกาศนโยบายต่างๆ ของภาครัฐเช่น นโยบาย Thailand 4.0 การนำเทคโนโลยีสารสนเทศและนวัตกรรมมาพัฒนามูลค่าและคุณค่า (Value Creation) เพื่อขับเคลื่อนเศรษฐกิจไทยไปสู่การปรับเปลี่ยนโครงสร้างเศรษฐกิจไปสู่ “Value-Based Economy” หรือ “เศรษฐกิจที่ขับเคลื่อนด้วยนวัตกรรม” และโดยเฉพาะงานด้านบริการต่างๆ จำเป็นต้องใช้เทคโนโลยีมาช่วยเป็นอย่างยิ่งเพราะต้องการความถูกต้อง รวดเร็ว ให้ทันกับความต้องการของผู้มารับบริการ ไม่เว้นแม้กระทั่งสถานที่ให้บริการด้านการแพทย์ อย่างคณะแพทยศาสตร์ศิริราชพยาบาล มหาวิทยาลัยมหิดล ที่ทำงานของผู้เขียนเอง ก็เป็นที่ยอมรับว่าฝ่ายสารสนเทศ

เป็นกระดูกสันหลังที่ขับเคลื่อนคณะแพทยศาสตร์ศิริราชพยาบาล มหาวิทยาลัยมหิดล ให้พันธกิจที่ประกาศไว้ประสบความสำเร็จ ซึ่งการใช้เทคโนโลยีสารสนเทศเข้ามาเป็นเครื่องมือในการให้บริการด้านการแพทย์นั้น ส่วนสำคัญอย่างหนึ่ง คือการทำงานบนข้อมูลขนาดใหญ่ เทคโนโลยีสารสนเทศที่ต้องทำงานร่วมกันนั้น ไม่ว่าจะเป็น Server, Network, Desktop Client, และอุปกรณ์อื่นๆ จะต้องทำงานร่วมกันได้อย่างมีประสิทธิภาพ รวมถึง Software ที่เลือกมาใช้ในแต่ละระบบงานก็ต้องเหมาะสมและฐานข้อมูลที่เลือกมาทำงานกับระบบงานเองก็ต้องเหมาะสมกับลักษณะของระบบงานด้วย และในส่วนของฐานข้อมูลนี้เองที่เป็นส่วนสำคัญ ที่จะส่งผลกับเรื่องการให้บริการที่ถูกต้อง รวดเร็ว มีความปลอดภัยของข้อมูลบนข้อมูลขนาดใหญ่มากที่สุด ซึ่งเทคโนโลยีฐานข้อมูลที่ถูกพัฒนาขึ้นเพื่อตอบโจทย์เรื่องการทำงานดังกล่าว คือ ฐานข้อมูล NewSQL ซึ่งถูกพัฒนาเพิ่มขึ้นจากฐานข้อมูล RDBMS (SQL) และ ฐานข้อมูล NoSQL

1. ประเภทของฐานข้อมูล (Type Of Databases)

ประเภทของฐานข้อมูลนั้นค่อนข้างจะแบ่งประเภทได้ยาก แต่ผู้เขียนจะขอแบ่ง Database โดยใช้ Standard Query Language (SQL) (พลชัย พิทักษานนท์กุลม, 2018) เป็นเกณฑ์ ซึ่งแบ่งออกเป็น 3 ประเภทหลักๆ โดยคือ ฐานข้อมูล RDBMS (SQL), ฐานข้อมูล NoSQL (พลชัย พิทักษานนท์กุลม, 2018) และ ฐานข้อมูล NewSQL (Rabi Prasad Padhy, 2018) และจะได้เปรียบเทียบทั้ง 3 ประเภทให้ทราบในลำดับต่อไปด้วย

1.1. Relational Databases

เริ่มต้นสร้างขึ้นมาจากกลุ่มของ Table ที่มีข้อมูลภายใน โดยแบ่งออกเป็นประเภทตามที่ตั้งไว้ใน Columns แต่ละ Table จะมีอย่างน้อย Column ในแต่ละ Row จะมีข้อมูลตามชนิดที่ Columns ได้กำหนดไว้ และมี Standard Query Language (SQL) เป็นภาษามาตรฐานที่ผู้ใช้งาน หรือระบบอื่นๆ เชื่อมต่อกับ Relational Databases ซึ่งง่ายต่อการเพิ่มข้อมูลเข้าไปใน Table โดยไม่กระทบต่อโปรแกรมอื่นที่ใช้งานร่วมกันอยู่ ตัวอย่าง Relational Databases เช่น Oracle, MySQL, Microsoft SQL Server, PostgreSQL และ IBM Db2

1.2. NoSQL Databases

NoSQL Databases ถูกใช้ในรูปแบบที่เป็น การกระจายของข้อมูล (Distributed Databases) จึงมีประสิทธิภาพสูง สำหรับข้อมูลขนาดใหญ่ (Big Data) เพราะ Relational Databases ไม่ได้ถูกออกแบบมาให้รองรับข้อมูลขนาดใหญ่เหมือน NoSQL Databases จึงทำให้ NoSQL Databases นิยมใช้กับการวิเคราะห์ข้อมูลขนาดใหญ่ที่ไม่ค่อยมีรูปแบบตายตัว NoSQL Databases มี 5 ประเภทและมีตัวอย่างฐานข้อมูลดังนี้

- ตัวอย่าง ฐานข้อมูล NoSQL ประเภท Key-Value Store คือ Redis, Oracle NoSQL Databases, Voldemorte, Aerospike

และ Oracle Berkeley DB

ตัวอย่างฐานข้อมูล NoSQL ประเภท

Document Store คือ MongoDB, Azure Cosmos DB, CouchDB, MarkLogic และ OrientDB

- ตัวอย่างฐานข้อมูล NoSQL ประเภท Column Store คือ Google Cloud Bigtable, Cassandra, HBase, Accumulo และ Microsoft Azure Table Storage

- ตัวอย่างฐานข้อมูล NoSQL ประเภท Graph Databases คือ

Neo4j, Blazegraph และ OrientDB

- ตัวอย่างฐานข้อมูล NoSQL ประเภท Multi Model Databases คือ ฐานข้อมูล OrientDB ใช้ Graph Model กับ Document Model, ฐานข้อมูล ArangoDB ใช้ Key- Value Model, Document Model, และ Graph Model และ ฐานข้อมูล Virtuoso เป็น Relational Databases, Graph Model, และ Document Model

1.3. NewSQL Databases

แนวคิดที่มาของ NewSQL Databases เริ่มมาจากความต้องการขยาย Database ให้รองรับการทำงานที่มากขึ้น ต้องการการทำงานของ Database ที่ดีขึ้น ทั้งเรื่องการทำงานด้าน Analytics รวดเร็ว มีความถูกต้อง มีความมั่นคงปลอดภัยของข้อมูล และใช้งานง่ายไม่ซับซ้อนซึ่ง RDBMS (SQL) และ NoSQL ยังไม่ตอบปัญหาได้ครบถ้วนนั่นเอง จึงเกิดการพัฒนา NewSQL Databases และจะขอรายละเอียดทั้งคุณลักษณะและประเภทของ NewSQL Databases, ข้อดี - ข้อเสียของ NewSQL Databases รวมถึงแนวโน้มในการใช้งาน NewSQL Databases ไปกล่าวอย่างละเอียดในหัวข้อต่อไป

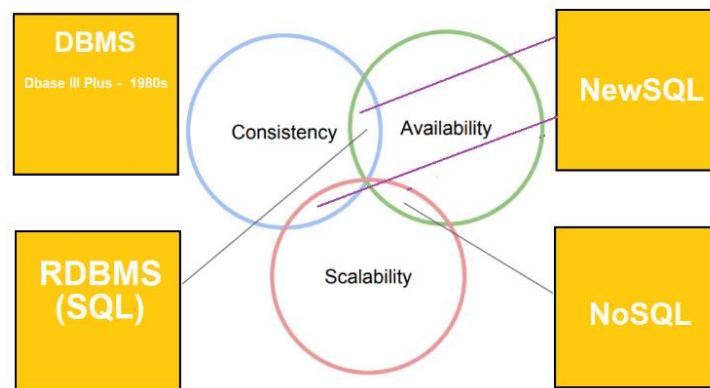
2. คุณสมบัติและประเภทของฐานข้อมูล NewSQL

ในยุคของการแปลง Digital Transformation บริษัทจำนวนมากย้ายข้อมูล Applications ไปยัง Public Cloud ฐานข้อมูล NewSQL ได้กลายเป็นจุดสนใจเนื่องจากมีคุณสมบัติสำคัญเช่น Horizontal Scalability, Support SQL และวิธีที่ยืดหยุ่นสำหรับผู้ใช้งานคือสามารถเข้าถึง

ข้อมูลโดยใช้ API Applications ที่เปิดใช้งานบนระบบ Cloud ต้องการฐานข้อมูลใหม่และ Engine ของ NewSQL ที่ออกแบบมาเพื่อรองรับ Applications ใหม่ ๆ และบริษัทส่วนใหญ่ย้ายออกจากแพลตฟอร์ม Data Warehouse แบบเก่าไปยัง “Data Lake” เพราะ Data Warehouse ไม่มีความยืดหยุ่นพอที่ Database ต้องการ ดังนั้นนวัตกรรมในปัจจุบันเกี่ยวกับฐานข้อมูล ที่กล่าวมา มั่นใจได้ว่าก้าวต่อไปของ Database จะเปลี่ยนไปทาง NewSQL Databases อย่างแน่นอน

คุณสมบัติของฐานข้อมูล NewSQL

คุณสมบัติของ NewSQL สามารถอธิบายให้เข้าใจง่ายๆ ด้วยรูปที่ 1



รูปที่ 1 คุณสมบัติของ NewSQL

Database ที่มีคุณสมบัติทำ Highly Scalable และ Distributed เรียกว่า NoSQL หรือ Database มีคุณสมบัติ Transactionally Consistent เรียกว่า RDBMS (SQL) ซึ่งคุณสมบัติเหล่านั้นได้อยู่ใน NewSQL ทั้งหมด และจะยกฐานข้อมูล Google Cloud Spanner มาอธิบายเพื่อให้เข้าใจคุณสมบัติ NewSQL ได้มากขึ้น

- คุณสมบัติที่สำคัญของฐานข้อมูล Google Cloud Spanner

ฐานข้อมูล Google Cloud Spanner (Ravi Prasad Padhy, 2018) ออกแบบมาเพื่อให้มีคุณสมบัติที่สามารถทำ Automatic Synchronously-Replicated, Globally-Distributed, Multi-Version Databases และมีคุณสมบัติที่สำคัญอื่นๆ ดังนี้

1. ฐานข้อมูล Google Cloud Spanner มีพร้อมใช้งานสูง (Highly available 99.999% SLA) เพราะใช้ Infrastructure ของ Google และเทคโนโลยี TrueTime ดังนั้นข้อมูลจึงมีพร้อมใช้งานสูงมาก
2. ฐานข้อมูล Google Cloud Spanner มี Externally Consistent โดยใช้ Two-Phase Commits และ Custom Paxos
3. ฐานข้อมูล Google Cloud Spanner มีการบริหารจัดการเรื่อง Replication และ Maintenance อย่างสมบูรณ์สามารถ Replication และ Maintenance อัตโนมัติได้เลย

4. ปรับขนาดได้อย่างง่ายดาย (Easily Scalable) ซึ่งเป็นข้อได้เปรียบที่สุดเพราะฐานข้อมูล Google Cloud Spanner รองรับแนวคิด Horizontal Scalability และออกแบบมาสำหรับ Horizontal Scalability สำหรับโหมด Read & Write ด้วย
5. ฐานข้อมูล Google Cloud Spanner รองรับ ACID Transactions
6. ฐานข้อมูล Google Cloud Spanner รองรับภาษา SQL
7. ฐานข้อมูล Google Cloud Spanner มี Security ระดับ Enterprise คือ การเข้ารหัสชั้นข้อมูล, การรวม IAM สำหรับการเข้าถึง และการควบคุม

จากคุณสมบัติข้างต้นที่กล่าวมาทำให้ฐานข้อมูล Google Cloud Spanner เป็นฐานข้อมูล NewSQL ที่เหมาะสำหรับองค์กรขนาดใหญ่ที่ต้องการความพร้อมทำงานตลอดเวลา (Highly available)

● Architecture Overview ของ Google Cloud Spanner

ฐานข้อมูล Google Cloud Spanner ได้รับการออกแบบเพื่อให้มีคุณสมบัติ Horizontal Scalability อย่างที่กล่าวมาแล้วและได้ถูกพัฒนาให้ทำงานง่ายกับ SQL Interface แต่ละ Deployed Spanner Instance จะเรียกว่า Universe หรือเรียกสั้นๆว่า Spanner Deployment

องค์ประกอบพื้นฐานของ Architecture ฐานข้อมูล Google Cloud Spanner มีดังนี้

1. Replica Consistency โดยใช้ Paxos Protocol
2. Concurrency Control โดยใช้ Two-Phase Locking
3. Transaction Coordination โดยใช้ Two-Phase Commit
4. Timestamps for Transactions โดยใช้ Multiple Timestamped Versions ที่มีการปรับปรุงรายการข้อมูล

● การ Deployment Google Cloud Spanner มีองค์ประกอบดังนี้

1. Universe Master ก็จะมี Master-process ทำงานประสานงานกับ Spanner Zone. Universe Master จะแสดงสถานะของ Zone

2. Spanner Zones คือส่วนที่มีไว้ให้ดูแลระบบ เช่น Deployment และทำ Cluster ซึ่งในแต่ละ Zone ประกอบด้วย Server หลายๆ เครื่อง Zones จะกระจายไป Datacenters อื่น ๆ ด้วย โดยในหนึ่ง Zone จะมีหนึ่ง Zone Master และสามารถมี Spanservers 100 ถึง 1,000 Spanservers

3. ZoneMaster (ต่อโซน): ZoneMasters จะทำหน้าที่กำหนดข้อมูลให้กับ Spanservers และ Server Clients ด้วย

4. Paxos เป็น Consensus Protocol ประเด็นสำคัญ คือการ Replication Database ในระดับ Global Scale นั้น ต้องให้ความสอดคล้องของข้อมูล แม้ว่าผู้ใช้จะทำ Multiple Transactions ซึ่งจำเป็นต้องเชื่อมต่อกับ Datacenter ระดับ Global ทั่วโลก ฐานข้อมูล Google Cloud Spanner จัดการคอขวดนั้นโดยใช้ระบบ Paxos ซึ่งอาศัย Algorithm ในการสร้าง Consensus ใน Nodes ที่กระจายไปทั่วโลกเพื่อกำหนดตำแหน่ง Addresses ที่จะทำการ Replicate ซึ่งส่วนใหญ่ขึ้นอยู่กับ Data gravity จะทำให้รู้ว่าใครเป็นผู้ที่จะต้อง Commit

5. Spanserver เป็นผู้ให้บริการข้อมูลแก่ Clients แต่ละ Spanserver จะรับผิดชอบ Client 100 ถึง 1,000 ที่ถือ Key-Value อยู่ และข้อมูลสามารถถูก Replicated ข้าม Spanservers ได้

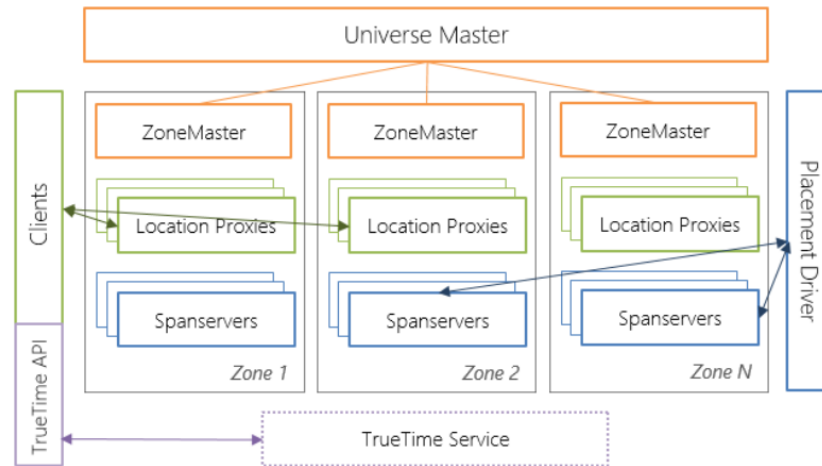
6. Location Proxy จะช่วยให้ Clients ค้นหา Spanservers ได้

7. Placement Driver เป็นกระบวนการที่ควบคุมการเคลื่อนไหวของข้อมูลระหว่างโซนที่ต่างกัน

8. TrueTime- นี่คือการใช้นาฬิกาแบบซิงโครไนซ์โดยใช้ตัวรับสัญญาณ GPS และนาฬิกาอะตอมมิกใน Datacenter ทุกแห่ง TrueTime ช่วยให้

ฐานข้อมูล Google Cloud Spanner มีส่วนสำคัญของข้อมูลสำหรับกิจกรรมการเขียน เช่น ระบบสามารถกำหนดลำดับการเขียนอย่างต่อเนื่อง แม้ใน Nodes ที่กระจายอยู่ที่ต่างๆ การใช้ข้อมูลที่กระจายอยู่ที่ต่างๆ

นั้นจะเกิดปัญหา Consistency ระหว่าง Nodes ใน Cluster ขึ้น ซึ่งสามารถจัดการกำหนดลำดับและป้องกันได้ด้วยการใช้ TrueTime API Sync



รูปที่ 2 Google Cloud Spanner Deployment

ฐานข้อมูล Google Cloud Spanner รองรับ Read-Write Transactions, Read-only Transactions และ Snapshot Reads การเขียนแบบ Standalone ถูกนำมาใช้เป็น Read-Write Transactions, Non-Snapshot Standalone Reads นำมาใช้เป็น Read-only Transactions, Snapshot Read นำมาใช้เป็นการอ่านข้อมูลในอดีต การควบคุม Concurrency ที่เกิดขึ้น โดยใช้ Two-Phase Locking ทำงานประสานกันกับ Two-Phase Commit การสร้างฐานข้อมูล Cloud Spanner นั้นง่ายมาก โดยใช้ Google Cloud Console และเพียงแค่ตั้งค่า ระบุจำนวนโหนดที่ต้องการและเลือกตำแหน่งภูมิภาคเท่านั้นเอง

ประเภทของฐานข้อมูล NewSQL

(Type of NewSQL Databases)

ฐานข้อมูล NewSQL มี 4 ประเภท ดังนี้

2.2.1. Historical Databases

ฐานข้อมูล Historical จะเก็บข้อมูลไว้ในระบบ Warehouse ขนาดใหญ่เมื่อเทียบกับ OLTP (Online Transaction Processing) (Ahmed

Almassabi, Omar Bawazeer and Salahadin Adam, 2018). ข้อมูลประวัติเป็นประโยชน์ในการช่วยทำนายอนาคต เมื่อทำการวิเคราะห์เชิงทำนายผล

เช่น ฐานข้อมูล AltiBase สามารถรองรับปริมาณงานจำนวนมากพร้อมทั้งทำการอนุญาตแบบ Real-Time ให้ผู้ใช้งานเข้าถึงข้อมูลสำคัญเมื่อจำเป็น การเข้าถึงข้อมูลประวัติและธุรกรรมที่ซับซ้อนเพื่อรวบรวมข้อมูล ด้วยการเคลื่อนย้ายข้อมูลแบบ Bidirectional ระหว่าง Disk และ Memory นอกจากนี้มีฐานข้อมูล C-TreeACE ให้ฟังก์ชัน Replication และการสำรองข้อมูลแบบ Hot Backup รวมกันมาเป็นแพ็คเกจเลย

2.2.2. Real-Time Databases

ฐานข้อมูล Real-Time เกี่ยวข้องกับการประมวลผลข้อมูล เพื่อระบุปริมาณงานของระบบและปริมาณการใช้ข้อมูลจำนวนมาก ซึ่งมีการปรับปรุงเปลี่ยนแปลงอยู่ตลอดเวลา มีกระบวนการสำคัญที่เรียกว่า Real-Time Processing ซึ่งคือการจัดการปริมาณงานที่มีการเปลี่ยนแปลงสถานะอยู่ตลอดเวลา Real-Time Databases (Ahmed Almassabi, Omar Bawazeer and Salahadin Adam, 2018). นี่แตกต่าง

จากฐานข้อมูลดั้งเดิม ซึ่งส่วนใหญ่ข้อมูลจะไม่ได้รับผลกระทบตามเวลา กลุ่มธุรกิจธนาคารและระบบติดตามการจัดส่งออนไลน์เหมาะสำหรับใช้ Real-Time Databases เพื่อแสดงให้เห็นถึงฐานข้อมูลประเภทนี้จะยกตัวอย่างง่ายๆเช่น เครื่องบินไม่ควรลงจอดก่อนที่จะถึงข้อมูล Real-Time บางอย่างเช่น น้ำมัน ความสูงและความเร็ว ดังนั้นการหน่วงเวลาสำหรับรอข้อมูลเหล่านี้ สำคัญมากไม่เช่นนั้นอาจนำไปสู่การที่จะเกิดอันตรายในการลงจอดได้

ฐานข้อมูล NewSQL แรกที่สนับสนุนฐานข้อมูลแบบ Real-Time คือ ฐานข้อมูล ClusterixDB จัดการปริมาณธุรกรรมจำนวนมากด้วยระบบการจัดการในตัว และการวิเคราะห์ข้อมูลการดำเนินงานของลูกค้าแบบ Real-Time ฐานข้อมูล MemSQL เข้มงวดมากในการประมวลผลการเข้าถึงข้อมูลแบบ Real-Time เพื่อสร้างการตัดสินใจสำหรับการเลือกข้อมูลที่อัปเดตหรือจะเพิกเฉยในการ Replicate สำหรับธุรกรรมที่สำคัญ

2.2.3. Streaming Databases

ฐานข้อมูล Streaming (Ahmed Almassabi, Omar Bawazeer and Salahadin Adam, 2018) เป็นข้อมูลที่สร้างขึ้นอย่างต่อเนื่องผ่านแหล่งข้อมูลจำนวนมาก โดยพิจารณาการชิงโครโนซีในขณะที่ข้อมูลยังมีเล็กน้อยขนาดไปบันทึก และ Streaming ข้อมูลที่เกี่ยวข้องกับข้อมูลที่หลากหลายเหล่านั้นด้วย เช่นไฟล์ Log

ตัวอย่างเช่น มีการทำแผนที่ทางภูมิศาสตร์ โดยใช้ฐานข้อมูล NuoDB เพื่อส่งข้อมูลโดยให้ผลลัพธ์สำหรับการรับส่งข้อมูลที่มีคุณภาพสูง NuoDB ไม่เก่งเรื่องการจัดการและจัดเก็บข้อมูล แต่มีประสิทธิภาพในการจัดการ Streaming ข้อมูลให้ทันสมัยอยู่เสมอเพื่อการจัดการและการสืบค้น, ฟีเจอร์ Cloud DevOps ของฐานข้อมูล ClusterixDB ช่วยให้สามารถปรับแต่งฐานข้อมูลได้อย่างเหมาะสมเพื่อจัดการกับปริมาณงานที่ Streaming ได้อย่างราบรื่นโดยไม่ต้องเปลี่ยนฮาร์ดแวร์หรือการเปลี่ยน Code ใด ๆ และอีกฐานข้อมูล NewSQL ที่สนับสนุนข้อมูล Streaming คือ ฐานข้อมูล MemSQL ซึ่งจัดการข้อมูล Streaming เพื่อควบคุมข้อมูลที่สำคัญ โดยมีการตรวจสอบความเหมาะสมในการเปลี่ยนชุดข้อมูล ฐานข้อมูล MemSQL อนุญาตให้มีการเติบโตของข้อมูล

ผ่านการจัดเก็บ In-Memory Row และ Disk-Based Column ความสามารถที่แตกต่างอีกอย่างของ MemSQL คือ Low Latency

2.2.4. Timestamp Databases

ตัวอย่างเช่น แอปพลิเคชันที่ใช้ฐานข้อมูล Google Cloud Spanner ก่อนจะเขียนต้องเปิดอ่านเวลา Timestamp (Ahmed Almassabi, Omar Bawazeer and Salahadin Adam, 2018) ก่อนและแอปพลิเคชันจะไม่สามารถเขียนไฟล์ที่ Timestamp เก่าได้ จะสามารถอ่านข้อมูลได้อย่างเดียวได้ยกตัวอย่าง Google Cloud Spanner ไว้อย่างละเอียดแล้วในหัวข้อ คุณสมบัติของฐานข้อมูล NewSQL

3. เปรียบเทียบฐานข้อมูล RDBMS (SQL), NoSQL and NewSQL

ฐานข้อมูล NewSQL มีคุณสมบัติที่รวมข้อดีของ ทั้ง RDBMS (SQL) และ NoSQL ไว้คือ

1. เรื่อง Schema : ฐานข้อมูล NewSQL สามารถทำได้ทั้ง Relational Schema แบบ RDBMS (SQL) และ Schema-free แบบ NoSQL
2. เรื่อง Scalability : ฐานข้อมูล NewSQL ก็ สามารถทำ Scalable Write/Read และ Horizontal Scalable แบบฐานข้อมูล NoSQL ซึ่งทางฐานข้อมูล RDBMS (SQL) ทำได้แค่ Scalable Read
3. เรื่อง High Availability : ฐานข้อมูล NewSQL ก็ทำการ Built-in High Availability มาให้ตั้งแต่แรกไม่ต้องมาทำการปรับแต่งเหมือนฐานข้อมูล RDBMS (SQL)
4. เรื่อง Storage : ก็สามารถทำงานได้ทั้ง On-Disk และ In-Memory เช่นเดียวกับฐานข้อมูล RDBMS (SQL) และฐานข้อมูล NoSQL (ถ้าฐานข้อมูลที่รองรับการทำงาน In-Memory อย่างเดียวก็จะมีปัญหาเรื่องการรองรับขนาดของข้อมูล)
5. เรื่อง Cloud Support : ฐานข้อมูล NewSQL เป็น Fully Supported ทำได้ดีกว่า ทั้งฐานข้อมูล RDBMS (SQL) และฐานข้อมูล NoSQL

6. เรื่อง Query Complexity : ฐานข้อมูล NewSQL มี Query เป็น Very High Complexity ซึ่งดีกว่าทั้งฐานข้อมูล NoSQL ที่ Query เป็น High Complexity และฐานข้อมูล RDBMS(SQL) ที่ Query เป็น Low Complexity

7. เรื่อง ACID-CAP-BASE : ฐานข้อมูล NewSQL เป็น ACID ก็ทำได้ดีแบบเดียวกับฐานข้อมูล RDBMS (SQL)

8. เรื่อง OLTP : ฐานข้อมูล NewSQL เป็น Fully Supported OLTP ทำได้ดีกว่าฐานข้อมูล RDBMS (SQL) ส่วนฐานข้อมูล NoSQL ไม่ Supported OLTP

9. เรื่อง Performance Overhead : ฐานข้อมูล NewSQL มี Overhead ขนาดเล็กที่สุด ซึ่งฐานข้อมูล NoSQL ที่มี Overhead ขนาดกลาง และฐานข้อมูล RDBMS (SQL) มี Overhead ขนาดใหญ่ที่สุด

10. เรื่อง Security Concerns : ข้อนี้ฐานข้อมูล NewSQL มี Security ต่ำเหมือนกับฐานข้อมูล NoSQL ซึ่งฐานข้อมูล RDBMS (SQL) มี Security สูงมาก เรื่อง Security เป็นข้อด้อยข้อเดียวจากคุณสมบัติข้อที่กล่าวมาทั้งหมด

4. ข้อดี - ข้อเสียของ NewSQL

Databases

นอกจาก ข้อดี - ข้อเสีย ที่ได้จากตารางเปรียบเทียบฐานข้อมูลในหัวข้อที่ 3 แล้ว จะสรุป ข้อดี-ข้อเสีย เพิ่มเติมของฐานข้อมูล NewSQL อีกดังนี้

3.1. ข้อดี

1. ลดความซับซ้อนของแอปพลิเคชันให้น้อยลง มีความมั่นคงที่แข็งแกร่งและสนับสนุนการทำธุรกรรมอย่างเต็มรูปแบบ
2. ใช้ SQL ที่คุ้นเคยและเครื่องมือมาตรฐานที่สะดวกกว่าฐานข้อมูล NoSQL
3. การวิเคราะห์ที่ดียิ่งขึ้นใช้ประโยชน์จาก SQL และส่วนขยาย

4. มีระบบ Clustering แบบฐานข้อมูล NoSQL ที่มีมากกว่า Data และ Query แบบดั้งเดิมของฐานข้อมูล RDBMS (SQL)

3.2. ข้อเสีย

1. ไม่มีฐานข้อมูล NewSQL ใดที่ใช้งานได้ง่ายเท่าฐานข้อมูล RDBMS (SQL)
2. เนื่องจากฐานข้อมูล NewSQL ส่วนใหญ่จะทำงานแบบ In-memory Architectures ปัญหาที่คือสถาปัตยกรรม In-memory (JOHN PIEKOS. , 2015) อาจไม่เหมาะสมสำหรับการทำงานบนข้อมูลขนาด Terabytes
3. ยังมี Tool เข้าถึงบริหารจัดการฐานข้อมูลน้อย เมื่อเทียบกับฐานข้อมูล RDBMS (SQL) ที่มีใช้กันมากมาย

4. ลักษณะงานที่เหมาะสมจะใช้ NewSQL

Databases

จริงๆแล้ว NewSQL Databases นั้นสามารถใช้งานได้กับทุกกลุ่มธุรกิจ หรือทุกกลุ่มงานก็ได้ เนื่องจากมีคุณลักษณะที่เป็นทั้ง RDBMS (SQL) และ NoSQL รวมอยู่ แต่กลุ่มธุรกิจที่ใช้งาน NewSQL Databases มากกว่ากลุ่มอื่นในปัจจุบันคือ Gaming, E-Commerce, Telecommunications industry หรืออื่นๆ ที่เน้น High Availability จะยกตัวอย่างลูกค้าของ NewSQL Databases CockroachDB, VoltDB, ClustrixDB, Google Cloud Spanner ดังนี้

5.1. ตัวอย่างลูกค้าที่ใช้ CockroachDB

- Nubank ย้ายบริการการอนุมัติบัตรเครดิตจากที่เก็บใน In-memory Storage มาใช้ CockroachDB เพื่อให้ได้โครงสร้างพื้นฐานที่ปรับขนาดได้ตลอดเวลาและการดำเนินงานที่มีประสิทธิภาพมากขึ้น
- Bose ใช้ CockroachDB เพื่อเพิ่มพลัง PaaS สำหรับผลิตภัณฑ์ใหม่ๆ ที่เชื่อมต่อกับระบบ Cloud

<http://doi.org/10.14456/jmu.2021.17>

- Rubrik ผู้นำตลาดด้านการจัดการข้อมูลบนระบบ Cloud กำลังสร้าง Metadastore ที่ไม่มีความผิดพลาดและมีความสม่ำเสมอสูงโดยใช้ CockroachDB
- MetroNOM ใช้ CockroachDB เพื่อเพิ่มประสิทธิภาพการให้บริการความพร้อมใช้งานสูงเพื่อรองรับพนักงานของบริษัท MetroAG เป็นบริษัท ขยายส่งที่มีพนักงาน 150,000 คนทำงานใน 32 ประเทศ
- Elliot กำลังใช้ CockroachDB เพื่อเพิ่มประสิทธิภาพให้กับแพลตฟอร์มชำระเงินอีคอมเมิร์ซทั่วโลกที่ใช้บริการของ Elliot
- Heroic Labs ลดความยุ่งยากใน Deployment การปรับปรุงเกมส์แบบกระจายด้วยบริการแบบ as-a-service ของ CockroachDB

5.2. ตัวอย่างลูกค้าที่ใช้ VoltDB

- Openet ต้องใช้ข้อมูลที่รวดเร็วในการลด TCO และรองรับข้อมูลมีสื่อแบบ Real-Time. Openet ใช้ VoltDB เป็นตัวเลือกสำหรับทำ Cloud-Deployable เพื่อทำ Transactional Database ที่สามารถจัดการกระแสข้อมูลปริมาณสูงได้อย่างยืดหยุ่นเพื่อบริษัทผู้ให้บริการสื่อสารตรวจสอบข้อมูลและใช้ประโยชน์แบบ Real-Time (Openet เป็นผู้ให้บริการชั้นนำสำหรับระบบสนับสนุนธุรกิจดิจิทัลแก่ บริษัทสื่อสารมือถือขนาดใหญ่รวมถึง T-Mobile, Sprint, Bell, Vodafone และอีกมากมาย)
- FT ลูกค้าคาดหวังว่าเว็บไซต์โปรดของพวกเขาจะตอบสนองทันทีหากพวกเขาต้องการและบริษัทที่ได้รับความไว้วางใจเหล่านี้ จะได้รับผลกำไรเป็นอย่างมาก หากพวกเขาตอบสนองได้อย่างที่ลูกค้าต้องการ FT.com เป็นเว็บไซต์สำหรับทำธุรกรรม Financial และ FT.com ต้องการทราบว่า มีอะไรที่มีอิทธิพลต่อพฤติกรรมของลูกค้า ณ เวลานั้นๆ และพวกเขาจำเป็นต้องเปลี่ยนหน้าเว็บไซต์ของผู้ใช้งาน โดยพิจารณาจากข้อมูลการ

คลิกล่าสุด ของคนคนนั้น เพื่อจูงใจผู้ใช้งานด้วยที่ต้องการใกล้คำว่า Real-Time และเพื่อเพิ่มความเป็นส่วนตัวของผู้ใช้งาน FT.com จึงเลือกใช้ฐานข้อมูล VoltDB

- Huawei เป็นผู้ให้บริการโซลูชันเทคโนโลยีสารสนเทศและการสื่อสารระดับโลกและเป็นยักษ์ใหญ่ด้านอุปกรณ์อิเล็กทรอนิกส์สำหรับผู้บริโภค Huawei เลือกฐานข้อมูล VoltDB ในการทำโซลูชันที่กำลังจะเปิดใช้ที่เรียกว่า FusionInsight Platform เป็นโซลูชันที่ใช้ในการวิเคราะห์ข้อมูลเพื่อการตรวจจับการฉ้อโกงทางการบริการด้านการเงิน และฐานข้อมูล VoltDB ยังได้รับการอนุมัติให้เป็นซัพพลายเออร์ทั่วทั้งองค์กรของ Huawei Technologies Co. Ltd
- DeltaDNA เป็นผู้ให้บริการวิเคราะห์ไลบรารีให้กับผู้พัฒนาเกมส์ และผู้พัฒนาเกมส์จะสามารถนำไลบรารีนั้นไปฝังไว้ในเกมส์ของพวกเขาเพื่อรับข้อมูลเชิงลึกที่สามารถดำเนินการใดๆต่อไป จากลักษณะงานที่ให้บริการแบบนี้ deltaDNA จึงต้องการระบบที่สามารถประมวลผลเหตุการณ์ที่ไม่จำกัดจำนวนและทำการตัดสินใจทันทีทันใดที่เกิดเหตุการณ์เหล่านั้น ซึ่ง deltaDNA เลือกใช้ฐานข้อมูล VoltDB

5.3. ตัวอย่างลูกค้าที่ใช้ ClustrixDB

- Samsung คุณ Kwangbock Lee หัวหน้า Database Architect ของ Samsung พูดไว้ในงาน MariaDB Open Works เมื่อ 25-27 กุมภาพันธ์ 2019 ที่ New York ว่า Samsung ใช้ฐานข้อมูล ClustrixDB ในการจัดการข้อมูลที่เติบโตอย่างรวดเร็วโดยไม่ต้องใช้ฐานข้อมูลด้วยตนเอง

5.4. ตัวอย่างลูกค้าที่ใช้ Google Cloud Spanner

- Streak สร้างฐานข้อมูลกราฟบน Google Cloud Spanner เพื่อรองรับอีเมลนับพันล้านฉบับที่ให้บริการ

- Bandai Namco Entertainment (BNE) ใช้ Google Cloud Spanner ทำงานเบื้องหลัง Scenes เกมส์ Dragon Ball Legends GCP
- Optiva ใช้ Google Cloud Spanner เปลี่ยนอุตสาหกรรมโทรคมนาคม
- Mixpanel ใช้ Google Cloud Spanner Scalable และใช้ในการวิเคราะห์ผลิตภัณฑ์อย่างมีประสิทธิภาพ

5. แนวโน้มในการใช้งาน NewSQL Databases

การที่จะตัดสินใจเลือกใช้ผลิตภัณฑ์ฐานข้อมูลกับระบบงานหรือแอปพลิเคชันนั้น สิ่งที่จะเป็นปัจจัยสำคัญก็คือ ประเภทของข้อมูลที่ใช้ งาน การออกแบบฐานข้อมูล ให้ความเหมาะสมกับลักษณะงาน รวมไปถึงความคุ้มค่าที่จะลงทุน โดยจะต้องศึกษากรณีศึกษา (Use Case) ชุมชน (Community) ต่างๆ และทดลองใช้งานจริง เพื่อให้มั่นใจว่าจะเลือกฐานข้อมูลได้เหมาะสมแล้ว

ซึ่งการจัดอันดับฐานข้อมูลทั้งหมด 10 อันดับแรก ของเว็บไซต์ DB- Engines แสดงดังรูปที่ 3

354 systems in ranking, March 2020									
Rank			DBMS	Database Model	Score				
Mar 2020	Feb 2020	Mar 2019			Mar 2020	Feb 2020	Mar 2019		
1.	1.	1.	Oracle +	Relational, Multi-model	1340.64	-4.11	+61.50		
2.	2.	2.	MySQL +	Relational, Multi-model	1259.73	-7.92	+61.48		
3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model	1097.86	+4.11	+50.01		
4.	4.	4.	PostgreSQL +	Relational, Multi-model	513.92	+6.98	+44.11		
5.	5.	5.	MongoDB +	Document, Multi-model	437.61	+4.28	+36.27		
6.	6.	6.	IBM Db2 +	Relational, Multi-model	162.56	-2.99	-14.64		
7.	7.	9.	Elasticsearch +	Search engine, Multi-model	149.17	-2.98	+6.38		
8.	8.	8.	Redis +	Key-value, Multi-model	147.58	-3.84	+1.46		
9.	9.	7.	Microsoft Access	Relational	125.14	-2.92	-21.07		
10.	10.	10.	SQLite +	Relational	121.95	-1.41	-2.92		

รูปที่ 3 Complete Ranking (Admin db-engines.com, 2020) by DB- Engines

จะเห็นได้ว่าจะมีเพียงพื้นฐาน RDBMS (SQL) และพื้นฐาน NoSQL ประเภทต่างๆที่คุ้นชื่อเท่านั้น ยังไม่มีฐานข้อมูล NewSQL ขึ้นมาในอันดับความนิยมเลย นั่นหมายถึงว่าไม่ง่ายเลยที่จะเป็นฐานข้อมูลที่ได้รับ ความนิยม

จากหัวข้อก่อนหน้านี้ได้ยกตัวอย่าง ลูกค้าของ NewSQL Databases ให้ได้ทราบแล้ว จะเห็นได้ว่าเริ่มมีการใช้งานกันมากพอสมควร เรียกว่าขยายไปหลายๆ กลุ่มธุรกิจ แต่ก็ยังเป็นองค์กรต่างประเทศทั้งหมด แต่ถ้าจะมองแนวโน้มการใช้งานจริงๆ ก็คงต้องพึ่งแหล่งข้อมูลที่มีการเก็บสถิติมาพอสมควร จึงจะขอยกข้อมูลของ db-engines.com มาแนะนำ ซึ่งเว็บไซต์ db-engines.com มีการให้คะแนนของแต่ละฐานข้อมูล และจัดอันดับความนิยมไว้

โดยคะแนนความนิยมจะเป็นค่าสัมพันธ์กับมูลค่าของผลิตภัณฑ์เสมอ ซึ่งควรตีความค่าความนิยมไว้เปรียบเทียบกับฐานข้อมูลอื่นเท่านั้น การจัดอันดับของเว็บไซต์ DB-Engines ไม่ได้วัดจำนวนการติดตั้งระบบหรือการใช้งานจริงภายในระบบไอที แต่สามารถใช้คาดการณ์แนวโน้มความนิยมล่วงหน้าได้ว่า การเพิ่มขึ้นของความนิยมของแต่ละฐานข้อมูล โดยใช้การจัดอันดับเว็บไซต์ DB-Engines เป็นตัวบ่งชี้ซึ่งสามารถใช้ในการอภิปรายหรือเสนองานได้

6.1. ข้อมูลแนวโน้มของฐานข้อมูลที่เป็นที่นิยมในปัจจุบัน

ก่อนที่จะดูแนวโน้มฐานข้อมูล NewSQL เราควรจะดูฐานข้อมูลที่เป็นที่นิยมในปัจจุบันที่ก่อนว่าแนวโน้มเป็นอย่างไร

6.1.1. Trend of Oracle Popularity

กำเนิดเมื่อปี 1980 เป็นอันดับที่ 1 ทั้ง Overall Databases และ Relational Databases ที่คะแนน 1340.64 ทั้งที่เป็น Commercial License แต่เมื่อดูจากกราฟแล้ว หลังปี 2014 มีแนวโน้มความนิยมลดลงทุกปี มีปี 2019 ที่ผ่านมารกราฟขึ้นพอสมควรแต่ก็ยังไม่ใกล้เคียงปี 2013 ที่ทำสูงที่สุดไว้ 1617.19 คะแนนและในปี 2020 นี้ทรงตัวที่ 1340 – 1350 คะแนน

6.1.2. Trend of Microsoft SQL Server Popularity

กำเนิดเมื่อปี 1989 เป็นอันดับที่ 3 ของทั้ง Overall Databases และ Relational Databases เป็น Commercial License และดูจากกราฟแล้วแนวโน้มความนิยมขึ้นสูงสุดในปี 2012 ที่ 1356.83 คะแนน และลดลงมาเรื่อยๆ จนกลางปี 2017 กราฟ เริ่มกลับขึ้นไปอีกครั้งที่ 1225.47 คะแนน และปัจจุบันปี 2020 มีความนิยมปรับลดลงมาพอๆกับปี 2015 ที่ 1097.85 คะแนน

6.1.3. Trend of MongoDB Popularity

กำเนิดเมื่อปี 2009 เป็นอันดับที่ 5 ของทั้ง Overall Databases เป็นอันดับที่ 1 ของ Overall NoSQL

Databases และเป็นอันดับที่ 1 ของ NoSQL Databases ประเภท Document Store เป็น Open Source License และดูจากกราฟแล้วแนวโน้มความนิยม เริ่มจากปี 2013 ที่ 96.12 คะแนน เพิ่มขึ้นต่อเนื่องในทุกปีถึงปัจจุบันปี 2020 มีคะแนนที่ 437.61 คะแนน

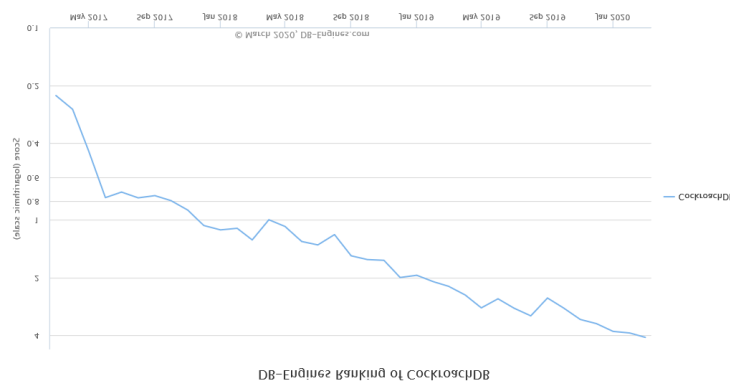
6.2. ข้อมูลพื้นฐานและกราฟแนวโน้มของ

ฐานข้อมูล NewSQL

มาดูข้อมูลพื้นฐานและกราฟแนวโน้มของฐานข้อมูล NewSQL ว่าเป็นอย่างไร (เว็บไซต์ DB-Engines มองฐานข้อมูล NewSQL เป็น ฐานข้อมูล แบบ Relational Databases เช่นเดียวกันกับพื้นฐาน RDBMS (SQL))

6.2.1. CockroachDB

กำเนิดเมื่อปี 2015 ฐานข้อมูล CockroachDB เป็นอันดับที่ 75 ของทั้ง Overall Databases และ เป็นอันดับที่ 40 ของ Relational Databases เป็น Open Source License และดูจากกราฟแล้วแนวโน้มความนิยม เริ่มจากปี 2017 เพิ่มขึ้นต่อเนื่องในทุกปีถึงปัจจุบันปี 2020 มีคะแนนที่ 4.06 คะแนน

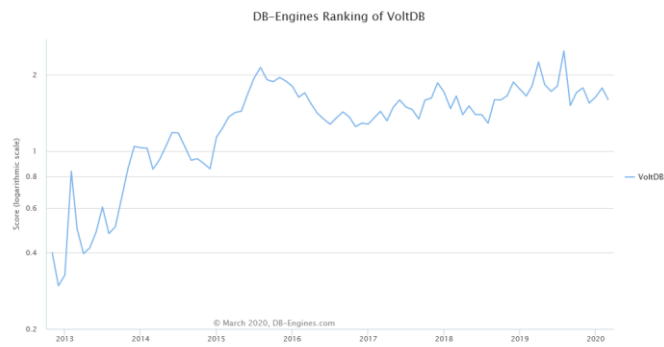


รูปที่ 4 Trend of CockroachDB Popularity

6.2.2. VoltDB

กำเนิดเมื่อปี 2010 ฐานข้อมูล VoltDB เป็นอันดับที่ 126 ของทั้ง Overall Databases และ เป็นอันดับที่ 63 ของ Relational Databases เป็น Open Source License และดู

จากกราฟแล้วแนวโน้มความนิยม เริ่มจากปี 2013 เพิ่มขึ้น และลดลงอยู่เป็นช่วงๆตลอด จนในปีปัจจุบันปี 2020 มีคะแนนที่ 1.60 คะแนนพอๆกับปี 2017

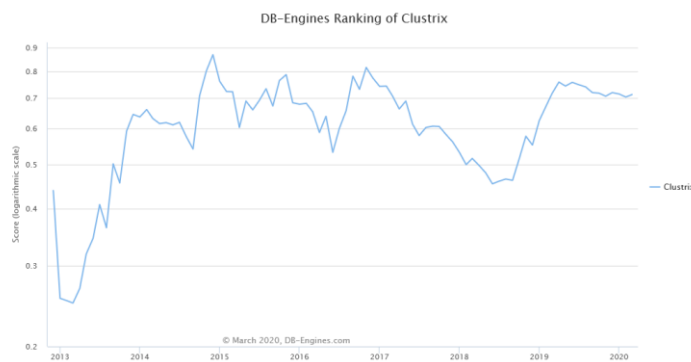


รูปที่ 5 Trend of VoltDB Popularity

6.2.3. ClustrixDB

กำเนิดเมื่อปี 2010 ฐานข้อมูล ClustrixDB เป็นอันดับที่ 175 ของทั้ง Overall Databases และ เป็นอันดับที่ 86

ของ Reational Databases เป็น Commercial License และดูจากกราฟแล้วแนวโน้มความนิยม เริ่มจากปี 2013 เพิ่มขึ้นต่อเนื่องในทุกปีถึงปัจจุบันปี 2020 มีคะแนนที่ 4.06 คะแนนพอๆกับปี 2016

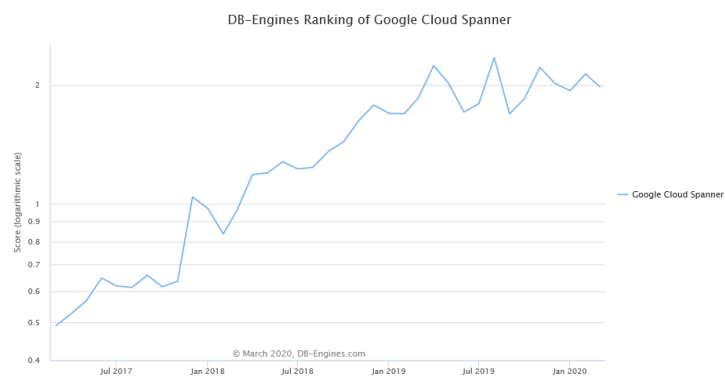


รูปที่ 6 Trend of Clustrix Popularity

6.2.4. Google Cloud Spanner

กำเนิดเมื่อปี 2017 ฐานข้อมูล Google Cloud Spanner เป็นอันดับที่ 112 ของทั้ง Overall Databases และ เป็นอันดับที่ 54 ของ Reational Databases เป็น

Commercial License และดูจากกราฟแล้วแนวโน้มความนิยม เริ่มจากปี 2017 เพิ่มขึ้นต่อเนื่องในทุกปีถึงปัจจุบันปี 2020 มีคะแนนที่ 1.98 คะแนน จะลดลงเล็กน้อยเมื่อเทียบกับปี 2019



รูปที่ 7 Trend of Google Cloud Spanner Popularity

จะเห็นได้ว่าฐานข้อมูลที่มีความนิยมนั้นไม่มีฐานข้อมูลไหนที่กำลังได้รับความนิยมแล้วได้รับความนิยมภายในเวลาอันสั้นต้องผ่านการพิสูจน์ความน่าเชื่อถือ โดยใช้ทั้งเวลาและการใช้งานจริงของลูกค้าในกลุ่มธุรกิจต่างๆ และเมื่อเกิดการเชื่อถือถึงจะแพร่เข้ามาติดอันดับความนิยมได้ แม้แต่ฐานข้อมูล NoSQL อย่างฐานข้อมูล MongoDB เอง การเป็น Open Source License ก็ไม่ใช่ปัจจัยหลักในการจะได้รับความนิยมอย่างรวดเร็วแต่อย่างใด ก็ต้องใช้เวลาพิสูจน์ความน่าเชื่อถือนานเกือบ 10 ปี เมื่อพิสูจน์ความน่าเชื่อถือได้แล้ว แม้จะเป็น Commercial License ก็ยังได้รับความนิยมต่อไปอย่างแน่นออย่าง เช่น Oracle, Microsoft SQL Server แม้ว่าดูจากกราฟความนิยมแล้วฐานข้อมูล NewSQL จะมีความนิยมเพิ่มขึ้นทุกปีแต่เมื่อเทียบกับคะแนนแล้วยังมีคะแนนนิยมน้อยมาก อาจจะมองได้ว่าเป็นความนิยมหรือความสนใจเฉพาะกลุ่มซึ่งอาจจะต้องใช้เวลามากหลายปีในการพิสูจน์ความน่าเชื่อถือ

ดังนั้น แนวโน้มการใช้งานของฐานข้อมูล NewSQL ขึ้นอยู่กับการสร้าง Use Case ให้มากที่สุด การสร้าง Community ให้ความรู้ให้มากที่สุด เพื่อให้เกิดความมั่นใจว่าเมื่อเกิดปัญหาใดๆ และจะหาทางแก้ไขปัญหานั้นได้ และจะต้องพิสูจน์ความน่าเชื่อถือให้ได้ในทุกด้านหรือน้อยให้เป็นที่ยอมรับได้ รวมถึงการบริการหลังการขายและความเชี่ยวชาญของทีมงานบริษัทเจ้าของผลิตภัณฑ์ด้วย

7. ความท้าทายในการใช้งาน NewSQL Databases

จะเห็นได้ว่า ตัวอย่างในหัวข้อต่างๆหรือตัวอย่างการใช้งานฐานข้อมูล NewSQL นั้น ทั้งหมดเป็นของประเทศอื่นๆที่ไม่ใช่ประเทศไทยเลย ความท้าทายแรกคือการจะมีบริษัทที่เกี่ยวข้องกับธุรกิจ Gaming, E-Commerce, Telecommunications industry (กลุ่มธุรกิจที่หันมาใช้ฐานข้อมูล NewSQL อย่างกว้างขวางพอสมควรในต่างประเทศ) บริษัทไหนในไทยที่จะกล้าลองใช้งานฐานข้อมูล NewSQL อย่างจริงจัง ความท้าทายที่สองที่ใกล้ตัวผู้เขียนเองคือการนำฐานข้อมูล NewSQL มาใช้งานในกลุ่มธุรกิจ Healthcare ในไทย ซึ่งในส่วนธุรกิจ Healthcare ก็จะมีฟังก์ชันงานมากมายที่จะนำฐานข้อมูล NewSQL ทุกประเภทมาทำการศึกษาการใช้งานหรือทดลองใช้งานได้ เช่น

- ฐานข้อมูล NewSQL ประเภท Historical Databases ตัวอย่างเช่น การเก็บประวัติข้อมูลผู้ป่วยในคลินิกโรคไตๆสักคลินิก แล้วนำข้อมูลที่เคยเก็บมานั้นมาประมวลผลเพื่อทำนายแนวโน้มของโรค กับคนที่ใช้ที่เข้ามาตรวจในคลินิกโรคนั้นๆ เพื่อการรักษาคนไข้ได้อย่างรวดเร็วและถูกต้องเหมาะสม (ซึ่งต้องใช้ข้อมูลจำนวนมากมาเป็นกรณีศึกษาเพื่อให้ได้ผลการทำนายแนวโน้มของโรคอย่างถูกต้องแม่นยำ รวมถึงต้องมีทีมแพทย์เข้ามาตรวจสอบความถูกต้องของการทำนายผลด้วย)

- ฐานข้อมูล NewSQL ประเภท Real-Time Databases ตัวอย่างเช่น การดึงข้อมูล Rel-Time จากอุปกรณ์ทางการแพทย์ในหอผู้ป่วยใน หรือห้องอุบัติเหตุ มาประมวลผลเพื่อประเมินอาการและแจ้งเตือนไปยังเจ้าหน้าที่เพื่อให้ผู้ป่วยได้รับการรักษาให้ทันท่วงทีก่อนที่ผู้ป่วยจะถึงขีดอันตราย

- ฐานข้อมูล NewSQL ประเภท Streaming Databases ตัวอย่างเช่น การ Tracking ผู้ป่วยว่าตอนนี้ผู้ป่วยเข้ารับการรักษาที่คลินิกไหน ขึ้นตอนใด หรืออยู่ส่วนไหนของโรงพยาบาล การ Tracking นั้นเป็นปัญหามากกับโรงพยาบาลที่มีบริเวณกว้างใหญ่ หรือมีหลายตึก หลายชั้น

- Timestamp Database ตัวอย่างเช่น การใช้งานในส่วนของการจัดการคิวผู้ป่วยเข้ารับการรักษา, คิวการจ่ายเงิน, คิวการรับยา ซึ่งขึ้นอยู่กับการ Timestamp ในแต่ละจุดบริการของผู้ป่วย

ตัวอย่างทั้งหมดที่ยกมานั้นไม่ได้หมายความว่าเหมาะสมหรือถูกต้องแต่อย่างใด การเลือกใช้ฐานข้อมูลก็จะขึ้นอยู่กับประเภทของข้อมูลที่ใช้ในการออกแบบ ว่าออกแบบให้เหมาะสมกับ Database ประเภทไหน ซึ่งอาจจะไม่ได้จำเป็นต้องใช้ฐานข้อมูล NewSQL เลยก็ได้ถ้าจะให้ยกตัวอย่างระบบที่มีอยู่ของคณะฯ แล้ว Migrate มาใช้ NewSQL Databases นั้นอาจจะลงรายละเอียดลำบากเพราะต้องเจาะลงไปถึงโครงสร้างของฐานข้อมูลที่มีอยู่ปัจจุบัน แต่ถ้าจะให้ยกตัวอย่างการขั้นตอนการ Migrate ฐานข้อมูล SQL มาใช้งาน NewSQL จะมีขั้นตอนหลักๆ ดังนี้ โดยจะเป็นการ Migrating (OLTP) จาก MySQL ไปยัง Google Cloud Spanner ซึ่ง Database อื่นๆ หลักการก็จะคล้ายกัน คือ

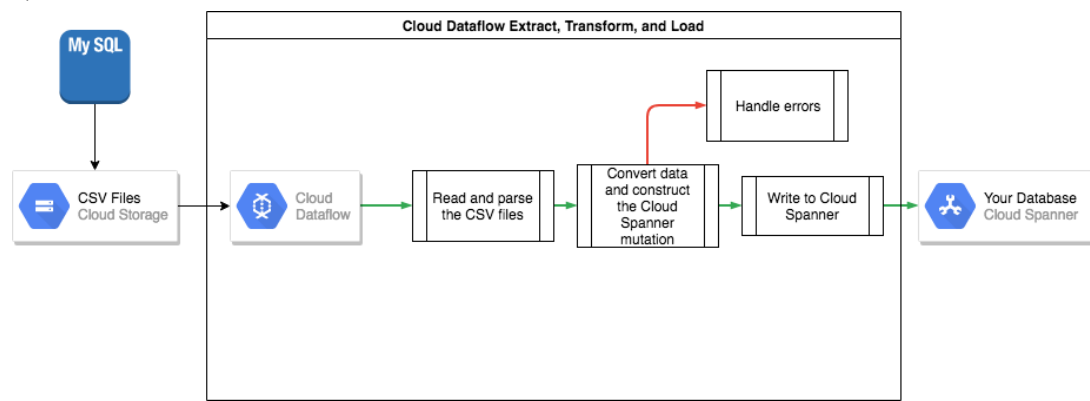
- ทำการ Convert Schema และ Data Model

Schema ที่มีอยู่ Convert เป็น Schema Google Cloud Spanner เพื่อเก็บข้อมูล ให้การปรับเปลี่ยนแอปพลิเคชันง่ายขึ้นตรวจสอบให้แน่ใจว่า Schema ที่ Convert นั้นตรงกับ Schema MySQL ที่มีอยู่หรือให้ใกล้เคียงที่สุด อย่างไรก็ตามเนื่องจากความแตกต่างของคุณสมบัติของฐานข้อมูล การ Convert มาอาจจะมีบางอย่างที่จำเป็นต้องมีการเปลี่ยนแปลง

- แปลง SQL Queries ต่างๆ โดย Google Cloud Spanner ใช้ภาษา ANSI 2011 ของ SQL โดยคำสั่ง SQL

ใด ๆ ที่ใช้ภาษาของ MySQL จะต้องถูกแปลงให้เข้ากันได้กับ Google Cloud Spanner

- Migrate แอปพลิเคชันเพื่อใช้ Google Cloud Spanner การ Migrate แอปพลิเคชัน ขั้นตอนสำคัญคือการถ่ายโอนข้อมูลจาก MySQL ไปยัง Google Cloud Spanner ซึ่งต้อง Export ฐานข้อมูล MySQL ไปเป็นรูปแบบ Portable File เช่น ไฟล์ XML, CSV แล้วนำเข้าข้อมูลนั้นไปยัง Google Cloud Spanner โดยใช้ Dataflow อ้างอิงตามรูปที่ 8



รูปที่ 8 Cloud Dataflow Extract, Transform, and Load

- Export ฐานข้อมูลจาก MySQL และ Import ฐานข้อมูลเข้า Spanner โดยใช้ Dataflow ใช้ mysqldump ที่มาพร้อมกับ MySQL สามารถ Export ฐานข้อมูลทั้งหมดมาเป็นไฟล์ XML เพื่อที่จะนำไปใช้ Import เข้า Google Cloud Spanner ได้ หรือคุณสามารถใช้คำสั่ง SELECT ... INTO OUTFILE SQL เพื่อสร้างไฟล์ CSV สำหรับแต่ละ Table ได้ แต่ข้อเสียคือทำได้ครั้งละหนึ่ง Table และในการส่งข้อมูลออกจะไม่สามารถใช้งาน Table นั้นได้ ซึ่งหมายถึงจะมีผลกับการใช้งานแอปพลิเคชัน หลังจากส่งออกไฟล์ข้อมูลเหล่านี้แล้วให้อัปโหลดไปยังที่ฝากข้อมูล Cloud Storage เพื่อให้สามารถนำเข้า Google Cloud Spanner ต่อได้สะดวกต่อไป

เนื่องจาก Schema ของฐานข้อมูลที่แตกต่างกันระหว่าง MySQL และ Google Cloud Spanner ซึ่งอาจจะบางขั้นตอนที่ต้องทำการแปลงข้อมูลในกระบวนการ Import วิธีที่ง่ายที่สุดในการแปลงข้อมูลเหล่านี้และนำเข้า

ข้อมูลไปยัง Google Cloud Spanner คือการใช้ Dataflow ซึ่ง Dataflow เป็นบริการแยกการแปลงและการโหลด (ETL) ของ Google Cloud โดยมีแพลตฟอร์มสำหรับการใช้งาน Data Pipelines ที่เขียนด้วย Apache Beam SDK เพื่ออ่านและประมวลผลข้อมูลจำนวนมากในแบบคู่ขนานกับเครื่องหลาย ๆ เครื่อง

- รักษาความสอดคล้องกันระหว่างฐานข้อมูล MySQL และ Google Cloud Spanner ระหว่างการย้ายข้อมูลแอปพลิเคชันจำนวนมากที่ต้องมีความพร้อมใช้งานตลอดเวลา ทำให้ไม่สามารถออฟไลน์แอปพลิเคชันในเวลาที่ต้องการ Export และ Import ข้อมูลได้ ซึ่งในขณะที่กำลังถ่ายโอนข้อมูลไปยัง Google Cloud Spanner แอปพลิเคชันจะยังทำงานอยู่และสามารถแก้ไขฐานข้อมูล MySQL ได้ อยู่ ดังนั้นจึงจำเป็นต้องทำการอัปเดตข้อมูล ให้ข้อมูลทั้งสองฐานข้อมูลให้เท่ากันก่อนจะย้ายแอปพลิเคชันในขั้นตอนสุดท้าย

- ขั้นตอนสุดท้าย แก้ไขแอปพลิเคชันที่เรียกใช้ MySQL
ให้ไปเรียกใช้ข้อมูลบนฐานข้อมูล Google Cloud Spanner

สรุป

จากที่กล่าวมาทั้งหมดจะเห็นได้ว่าไม่มีฐานข้อมูลใดหรือประเภทไหน ที่สามารถรองรับความต้องการหรือรองรับการใช้งานได้ดีที่สุดทุกด้าน ฐานข้อมูลแต่ละประเภทนั้นก็ยังมีข้อดีข้อเสียแตกต่างกันไป และฐานข้อมูลทุกประเภทนั้นก็ยังมีพัฒนาต่อยอดไปเรื่อยๆเช่นกัน ดังนั้นสิ่งที่ทุกคนในวงการ IT ควรทำ (และไม่จำกัดเฉพาะกับเรื่อง ฐานข้อมูล เท่านั้น) ก็คือต้องไม่หยุดการศึกษาเทคโนโลยีใหม่ๆ ศึกษาให้เข้าใจ ต้องมีการทดสอบการทำงานก่อนเลือกใช้งานจริง และเลือกใช้ประเภทของเทคโนโลยีให้เหมาะกับระบบงานของเรา เพราะถ้าหากเลือกเทคโนโลยีไม่เหมาะสมกับระบบงานแล้ว สุดท้ายก็จะก่อให้เกิดปัญหากับระบบงานในอนาคตอย่างแน่นอน

ส่วนแนวโน้มของการใช้งานฐานข้อมูล NewSQL นั้น ถึงแม้จะมีข้อเสียบ้างหรือมีค่าใช้จ่ายสูงเพียงใด สำหรับองค์กรขนาดใหญ่หรือองค์กรที่มีการแข่งขันทางด้านธุรกิจสูง หากการลงทุนเปลี่ยนแปลงองค์กรด้วยเทคโนโลยีหรือมีแนวทางใด ที่จะทำให้ได้เปรียบหรือนำหน้าคู่แข่งขึ้นเพียงเล็กน้อยมันก็น่าสนใจที่จะลงทุน การที่จะทำให้ผู้บริหารองค์กรเหล่านั้นหันมาใช้งาน ก็จะขึ้นอยู่กับการพิสูจน์ความน่าเชื่อถือของฐานข้อมูล NewSQL เอง ว่ามีความน่าเชื่อถือเพียงพอเป็นที่ยอมรับได้ ซึ่งต้องใช้เวลาและการใช้งานจริงของลูกค้าในกลุ่มธุรกิจต่างๆ เพื่อสร้างกรณีศึกษาให้มากที่สุด เพื่อเป็นการแสดงความเชี่ยวชาญในการแก้ปัญหาของเจ้าของผลิตภัณฑ์ การมี Community ให้ความรู้ เพื่อให้เกิดความมั่นใจว่าเมื่อเกิดปัญหาใดๆ และจะหาทางแก้ไขปัญหานั้นได้ โดยไม่ต้องอาศัยเจ้าของผลิตภัณฑ์อย่างเดียว

กิตติกรรมประกาศ

บทความทางวิชาการฉบับนี้จะสำเร็จไม่ได้หากไม่ได้รับการสนับสนุนจาก ฝ่ายสารสนเทศ คณะ

แพทยศาสตร์ศิริราชพยาบาล ผู้บังคับบัญชา และเพื่อนร่วมทีมทุกท่าน ที่แบ่งปันความรู้ และให้คำแนะนำต่างๆ ทำให้ผู้เขียนมีความรู้ความเข้าใจในเนื้อหาที่ปฏิบัติงานมีประสบการณ์ที่นำมาเขียนผลงานบทความทางวิชาการนี้ได้

เอกสารอ้างอิง

- พลชัย พิทักษานนท์กุล. (2018). ระบบฐานข้อมูล Database System. [เว็บไซต์]. สืบค้นเมื่อ 7 กุมภาพันธ์ 2019, จาก <http://www.glurgeek.com/education/ระบบฐานข้อมูล-database-system>
- Rabi Prasad Padhy. (2018). Google Spanner: A NewSQL Journey or Beginning of the End of the NoSQL Era. [เว็บไซต์]. เมื่อ 10 มกราคม 2019, จาก <https://medium.com/rabiprasadpadhy/google-spanner-a-newsq-journey-or-beginning-of-the-end-of-the-nosql-era-3785be8e5c38>
- Ahmed Almassabi, Omar Bawazeer and Salahadin Adam. (2018). TOP NEWSQL DATABASES AND FEATURES CLASSIFICATION. International Journal of Database Management Systems (IJDBMS) Vol.10, No.2, April 2018. [บทความ]. สืบค้นเมื่อ 10 ตุลาคม 2019, จาก <http://aircconline.com/ijdbms/V10N2/10218ijdbms02.pdf>
- JOHN PIEKOS. (2015). SQL VS. NOSQL VS. NEWSQL: FINDING THE RIGHT SOLUTION. [เว็บไซต์]. สืบค้นเมื่อ 01 ตุลาคม 2019, จาก <https://dataconomy.com/2015/08/sql-vs-nosql-vs-newsq-finding-the-right-solution/>
- Admin db-engines.com. (2020). DB-Engines Ranking. [เว็บไซต์]. สืบค้นเมื่อ 10 มีนาคม 2020, จาก <https://db-engines.com/en/ranking>