

NoSQL



I . NoSQL là gì?

1. Thuật ngữ

NoSQL còn có nghĩa là Non-Relational (NoRel) - không ràng buộc. Tuy nhiên, thuật ngữ đó ít phổ dụng hơn và ngày nay người ta thường dịch NoSQL thành **Not Only SQL - Không chỉ SQL**. NoSQL ám chỉ đến những cơ sở dữ liệu không dùng mô hình dữ liệu quan hệ để quản lý dữ liệu trong lĩnh vực phần mềm.

2. Lịch sử

Thuật ngữ NoSQL được giới thiệu lần đầu vào năm 1998 sử dụng làm tên gọi chung cho các lightweight open source relational database (cơ sở dữ liệu quan hệ nguồn mở nhỏ) nhưng không sử dụng SQL cho truy vấn.

Vào năm 2009, Eric Evans, nhân viên của Rackspace giới thiệu lại thuật ngữ NoSQL trong một hội thảo về cơ sở dữ liệu nguồn mở phân tán. Thuật ngữ NoSQL đánh dấu bước phát triển của thế hệ database mới: distributed (phân tán) + non-relational (không ràng buộc).



Ghi chú: *Một mệnh đề khá thú vị về non-relational data store: "select fun, profit from real_world where relational=false;"*.

3. Định nghĩa

Thế hệ database kế tiếp là một thế hệ cơ sở dữ liệu non-relational (không ràng buộc), distributed (phân tán), open source, horizontal scalable (khả năng mở rộng theo chiều ngang) có thể lưu trữ, xử lý từ một lượng rất nhỏ cho tới hàng petabytes dữ liệu trong hệ thống có độ chịu tải, lỗi cao với những đòi hỏi về tài nguyên phần cứng thấp.

Một số đặc điểm nhận dạng cho thế hệ database mới này bao gồm: schema-free, hỗ trợ mở rộng dễ dàng, API đơn giản, eventual consistency (nhất quán cuối) và/hoặc transactions hạn chế trên các thành phần dữ liệu đơn lẻ, không giới hạn không gian dữ liệu,...

NoSQL storage đặc biệt phổ dụng trong thời kỳ Web 2.0 bùng nổ, nơi các mạng dịch vụ dữ liệu cộng đồng cho phép người dùng tạo hàng tỷ nội dung trên web. Do đó, dữ liệu lớn rất nhanh vượt qua giới hạn phần cứng và cần phải giải quyết bằng bài toán phân tán. Nửa đầu năm 2009, người ta đã manh nha thuật ngữ NoSQL đánh dấu sự trưởng thành của thế hệ database mới trong khi những sản phẩm phần mềm có thể đã được phát triển từ trước đó rất lâu.

4. Một số thuật ngữ liên quan.

- non-relational: relational - ràng buộc - thuật ngữ sử dụng đến các mối quan hệ giữa các bảng trong cơ sở dữ liệu quan hệ (RDBMs) sử dụng mô hình khóa gồm 2 loại khóa: khóa chính và khóa phụ (primary key + foreign key) để ràng buộc dữ liệu nhằm thể hiện tính nhất quán dữ liệu từ các bảng khác nhau.

Non-relational là khái niệm không sử dụng các ràng buộc dữ liệu cho nhất quán dữ liệu ở NoSQL database.

- distributed storage: mô hình lưu trữ phân tán các file hoặc dữ liệu ra nhiều máy tính khác nhau trong mạng LAN hoặc Internet dưới sự kiểm soát của phần mềm.

- eventual consistency (nhất quán cuối): tính nhất quán của dữ liệu không cần phải đảm bảo ngay tức khắc sau mỗi phép write. Một hệ thống phân tán chấp nhận những ảnh hưởng theo phương thức lan truyền và sau một khoảng thời gian (không phải ngay tức khắc), thay đổi sẽ đi đến mọi điểm trong hệ thống, tức là cuối cùng (eventually) dữ liệu trên hệ thống sẽ trở lại trạng thái nhất quán.

- vertical scalable (khả năng mở rộng chiều dọc): Khi dữ liệu lớn về lượng, phương pháp tăng cường khả năng lưu trữ và xử lý bằng việc cải tiến phần mềm và cải thiện phần cứng trên một máy tính đơn lẻ được gọi là khả năng mở rộng chiều dọc. Ví dụ việc tăng cường CPUs, cải thiện đĩa cứng, bộ nhớ trong một máy tính,... cho DBMs nằm trong phạm trù này. Khả năng mở rộng chiều dọc còn có một thuật ngữ khác scale up.

- horizontal scalable (khả năng mở rộng chiều ngang): Khi dữ liệu lớn về lượng, phương pháp tăng cường khả năng lưu trữ và xử lý là dùng nhiều máy tính phân tán. Phân tán dữ liệu được hỗ trợ bởi phần mềm tức cơ sở dữ liệu.

Trong khi giá thành phần cứng ngày càng giảm, tốc độ xử lý, bộ nhớ ngày càng tăng thì horizontal scalable là một lựa chọn đúng đắn. Hàng trăm máy tính nhỏ được chập lại tạo thành một hệ thống tính toán mạnh hơn nhiều so với vi xử lý RISC truyền thống đơn lẻ. Mô hình này tiếp tục được hỗ trợ bởi các công nghệ kết nối Myrinet và InfiniBand. Từ đó chúng ta có thể quản lý, bảo trì từ xa, xây dựng batch procession (xử lý đồng loạt tập lệnh) tốt hơn. Do những đòi hỏi về tốc độ xử lý I/O cao, lượng cực lớn dữ liệu,... scale horizontally sẽ thúc đẩy các công nghệ lưu trữ mới phát triển giống như object storage devices (OSD).

II. Kiến trúc

1. Sơ lược.

Các RDBMs hiện tại đã bộc lộ những yếu kém như việc đánh chỉ mục một lượng lớn dữ liệu, phân trang, hoặc phân phối luồng dữ liệu media (phim, ảnh, nhạc, ...). Cơ sở dữ liệu quan hệ được thiết kế cho những mô hình dữ liệu nhỏ thường xuyên đọc viết trong khi các Social Network Services lại có một lượng dữ liệu cực lớn và cập nhật liên tục do số lượng người dùng quá nhiều ở một thời điểm. Thiết kế trên Distributed NoSQL giảm thiểu tối đa các phép tính toán, I/O liên quan kết hợp với batch processing đủ đảm bảo được yêu cầu xử lý dữ liệu của các mạng dịch vụ dữ liệu cộng đồng này. Facebook, Amazon là những ví dụ điển hình.

Về cơ bản, các thiết kế của NoSQL lựa chọn mô hình lưu trữ tập dữ liệu theo cặp giá trị key-value. Khái niệm node được sử dụng trong quản lý dữ liệu phân tán. Với các hệ thống phân tán, việc lưu trữ có chấp nhận trùng lặp dữ liệu. Một request truy vấn tới data có thể gửi tới nhiều máy cùng lúc, khi một máy nào đó bị chết cũng không ảnh hưởng nhiều tới toàn bộ hệ thống. Để đảm bảo tính real time trong các hệ thống xử lý lượng lớn, thông thường người ta sẽ tách biệt

database ra làm 2 hoặc nhiều database. Một database nhỏ đảm bảo vào ra liên tục, khi đạt tới ngưỡng thời gian hoặc dung lượng, database nhỏ sẽ được gộp (merge) vào database lớn có thiết kế tối ưu cho phép đọc (read operation). Mô hình đó cho phép tăng cường hiệu suất I/O - một trong những nguyên nhân chính khiến performance trở nên kém.

2. Một số đặc điểm.

- High Scalability: Gần như không có một giới hạn cho dữ liệu và người dùng trên hệ thống.
- High Availability: Do chấp nhận sự trùng lặp trong lưu trữ nên nếu một node (commodity machine) nào đó bị chết cũng không ảnh hưởng tới toàn bộ hệ thống.
- Atomicity: Độc lập data state trong các operation.
- Consistency: chấp nhận tính nhất quán yếu, cập nhật mới không đảm bảo rằng các truy xuất sau đó thấy ngay được sự thay đổi. Sau một khoảng thời gian lan truyền thì tính nhất quán cuối cùng của dữ liệu mới được đảm bảo.
- Durability: dữ liệu có thể tồn tại trong bộ nhớ máy tính nhưng đồng thời cũng được lưu trữ lại đĩa cứng.
- Deployment Flexibility: việc bổ sung thêm/loại bỏ các node, hệ thống sẽ tự động nhận biết để lưu trữ mà không cần phải can thiệp bằng tay. Hệ thống cũng không đòi hỏi cấu hình phần cứng mạnh, đồng nhất.
- Modeling flexibility: Key-Value pairs, Hierarchical data (dữ liệu cấu trúc), Graphs.
- Query Flexibility: Multi-Gets, Range queries (load một tập giá trị dựa vào một dãy các khóa).

3. What is NoSQL (technically speaking)?

Đừng gọi chúng là database. CTO của Amazon, Werner Vogels đề cập đến hệ thống Dynamo của họ đã gọi nó là một "highly available key-value store". Google gọi BigTable để nhấn mạnh đây là "distributed storage system for managing structured data" (hệ thống lưu trữ và quản lý dữ liệu cấu trúc có phân tán).

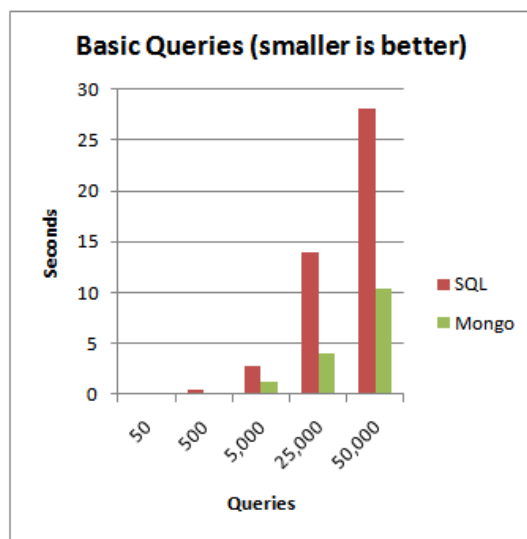
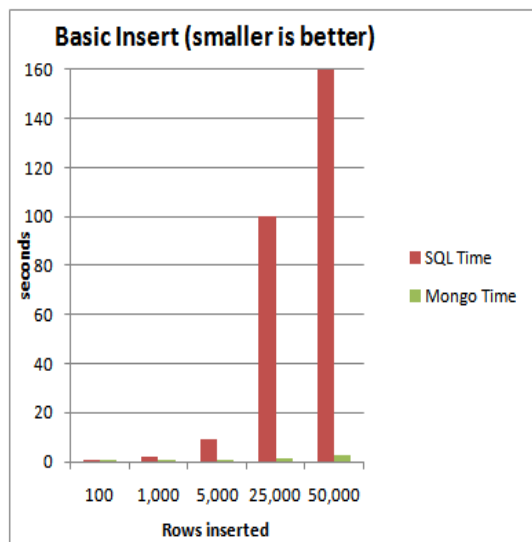
Có thể thổi bay một lượng dữ liệu cực lớn. Hypertable, một open source column-based database trên mô hình BigTable được sử dụng cho local search engine của Zvents Inc có thể ghi tới 1 tỷ cell dữ liệu mỗi ngày (theo Doug Judd một kỹ sư của Zvents). Trong khi đó BigTable kết hợp với MapReduce có thể xử lý tới 20 petabytes dữ liệu mỗi ngày.

Đánh bại performance bottlenecks. Bằng việc bỏ qua thông dịch trong SQL cùng với những truy vấn rườm rà, NoSQL cho ta một kiến trúc tối ưu về tốc độ thực thi (ghi và truy vấn dữ liệu).

Việc sử dụng các ràng buộc quan hệ cùng truy vấn SQL có vẻ thân thiện và thích hợp với phần đông dữ liệu. Tuy nhiên, nếu dữ liệu quá đơn giản, các thủ tục SQL sẽ không cần thiết (theo Curt Monash - một nhà phân tích cơ sở dữ liệu, một blogger).

Raffaele Sena, một senior computer scientist ở Adobe Systems Inc. đã nói rằng ConnectNow Web collaboration service của họ sử dụng Java clustering software từ Terracotta thay cho cơ sở dữ liệu quan hệ đã khiến "hệ thống của họ trở nên mạnh hơn, phức tạp hơn so với việc sử dụng cơ sở dữ liệu quan hệ".

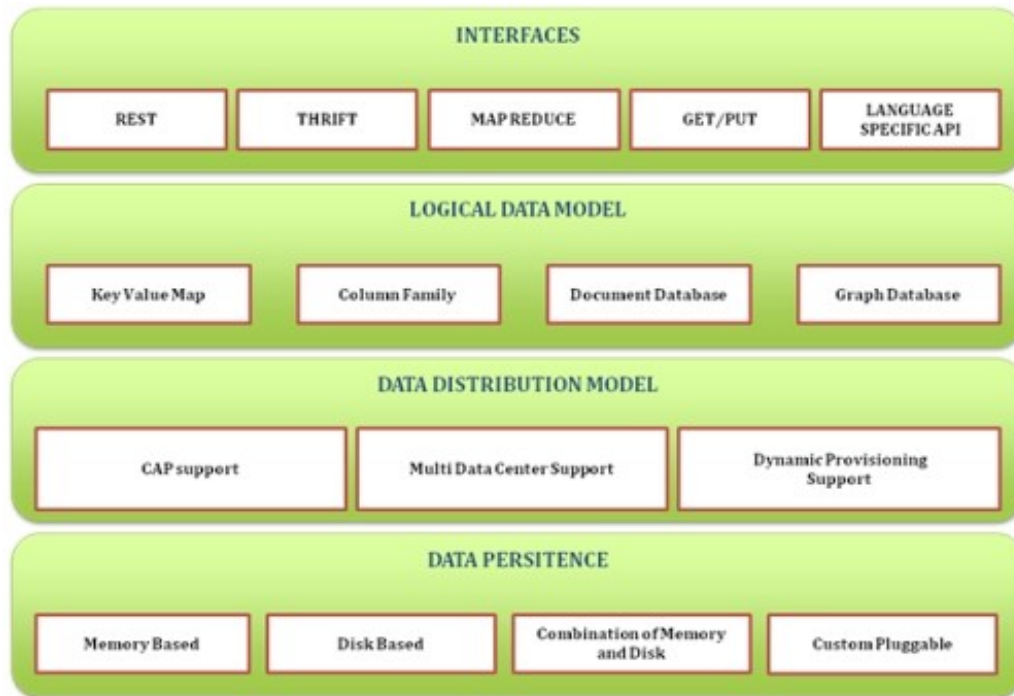
Các thiết kế database có tính đặc thù (như document-oriented database) sẽ lược bỏ được tầng chuyển đổi sang mô hình lưu trữ quan hệ từ interface của nó đồng thời khiến giao tiếp tương tác trở nên tự nhiên hơn.



MongoDB vs. SQL Server 2008 Performance Showdown

Không quá cần thiết. Đồng ý rằng RDBMs cung cấp một mô hình tuyệt vời để đảm bảo tính toàn vẹn dữ liệu. Tuy nhiên, rất nhiều người lựa chọn NoSQL đã nói rằng chúng không quá cần thiết cho nhu cầu của họ. Như trong dự án ConnectNow của Adobe, dữ liệu người dùng trong một session không cần thiết phải lưu lại, chúng sẽ bị xóa khi người dùng logoff. Vì vậy, một key-value memory storage là đủ dùng.

III. Phân loại



A. Core NoSQL Systems

1. Wide Column Store / Column Families

Hệ cơ sở dữ liệu phân tán cho phép truy xuất ngẫu nhiên/tức thời với khả năng lưu trữ một lượng cực lớn data có cấu trúc. Dữ liệu có thể tồn tại dạng bảng với hàng tỷ bản ghi và mỗi bản ghi có thể chứa hàng triệu cột. Một triển khai từ vài trăm cho tới hàng nghìn commodity hardware dẫn đến khả năng lưu trữ hàng petabytes nhưng vẫn đảm bảo high performance. Dưới đây là một số sản phẩm thông dụng: Hadoop/HBase – Apache, BigTable – Google, Cassandra - Facebook/Apache, Hypertable - Zvents Inc/Baidu, Cloudera, SciDB, Mnesia, Tablets,...

2. Key-Value Store/Tuple store

Mô hình lưu trữ dữ liệu dưới cặp giá trị key-value trong đó việc truy xuất, xóa, cập nhật giá trị thực thông qua key tương ứng. Với sự hỗ trợ bởi các kỹ thuật BTree, B+Tree, Hash,... dữ liệu có thể tồn tại trên RAM hoặc đĩa cứng, phân tán hoặc không phân tán. Hầu hết các NoSQL database đều là key-value store. Dưới đây là một số đặc tính có thể được hỗ trợ trong sản phẩm dạng này:

- Key/value cache in RAM: **memcached**, Citrusleaf database, Velocity, Redis, Tuple space,...
- Key/value save on disk: Memcachedb, **Berkeley DB**, Tokyo Cabinet, Redis,...

c. Eventually Consistent Key Value Store: **Amazon Dynamo**, Voldemort, Dynamite, KAI, **Cassandra**, Hibari, Project Voldemort,...

d. Ordered key-value store: NMDB, Memcachedb, **Berkeley DB**,...

e. Distributed systems: **Apache River**, MEMBASE, **Azure Table Storage**, **Amazon Dynamo**,...

Ghi chú: Tuple Store: schema free, an ordered list of elements.

3. Document Store

Thực chất là các document-oriented database - một thiết kế riêng biệt cho việc lưu trữ document. Các cài đặt có thể là giả lập tương tác trên relational database, object database hay key-value store. Một số sản phẩm tiêu biểu: **Apache Jackrabbit**, **CouchDB**, **IBM Lotus Notes Storage Format (NSF)**, **MongoDB**, Terrastore, ThruDB, OrientDB, RavenDB,...

4. Graph Database

Graph database là một dạng cơ sở dữ liệu được thiết kế riêng cho việc lưu trữ thông tin đồ họa như cạnh, nút, properties. Một số sản phẩm tiêu biểu: **Neo4J**, Sones, AllegroGraph, Core Data, DEX, FlockDB, **InfoGrid**, OpenLink Virtuoso,...

B. Soft NoSQL Systems

1. Object Databases: db4o, GemStone/S, InterSystems Caché, JADE, JOOB, Objectivity/DB, ZODB, Versant, Objectivity, NEO,...

2. XML Databases: Mark Logic Server, EMC Documentum xDB, eXist, Sedna, Berkeley DB XML, ...

3. Multivalue Databases: U2, OpenInsight, OpenQM

4. Grid Database Solutions: GigaSpaces, Hazelcast, GridGain, Infinispan, Coherence, ...

5. other NoSQL related databases: IBM Lotus/Domino, ISIS Family, Prevayler, eXtremeScal,...

V. Tài liệu tham khảo và lược dịch

1. NoSQL resources: <http://nosql-database.org/>

2. NoSQL wiki - <http://en.wikipedia.org/wiki/NoSQL>

3. Scalability wiki - http://en.wikipedia.org/wiki/Scalability#Scale_horizontally_28scale_out.29

4.No to SQL? Anti-database movement gains steam -

http://www.computerworld.com/s/article/9135086/No_to_SQL_Anti_database_movement_gains_steam_?taxonomyId=173&pageNumber=1

5. MongoDB vs. SQL Server 2008 Performance Showdown -

<http://www.michaelkennedy.net/blog/2010/04/29/MongoDBVsSQLServer2008PerformanceShowdown.aspx>

6. A Brief History of NoSQL - <http://blog.knuthaugen.no/2010/03/a-brief-history-of-nosql.html>

7. NoSql Databases – Part 1 - Landscape - <http://www.vineetgupta.com/2010/01/nosql-databases-part-1-landscape.html>

8. Nhất Quán Cuối Cùng - <http://www.sqlviet.com/blog/nhat-quan-cuoi-cung>

9. NoSQL in the Enterprise - <http://www.infoq.com/articles/nosql-in-the-enterprise>

Nhữ Đình Thuận