**Giao tác**

Aerospike là một cơ sở dữ liệu Khóa-Giá trị hiệu suất cao. Nó nhắm đến các tổ chức và trường hợp sử dụng cần thông lượng cao, với độ trễ thấp (hoàn thành 95% trong <1ms), đồng thời quản lý lượng lớn dữ liệu với 100% thời gian hoạt động, khả năng mở rộng và chi phí thấp.

Đối với một lớp dữ liệu, tính đúng đắn là rất quan trọng. Các giao tác cơ sở dữ liệu cung cấp đảm bảo ACID cho các dữ liệu đó:

* Atomic (Tính nguyên tố): Thành công hoặc thất bại về tổng thể.
* Consistency (Tính nhất quán): Để cơ sở dữ liệu ở trạng thái nhất quán.
* Isolation (Tính độc lập): Có thể được sắp xếp theo thứ tự đối với các giao tác khác.
* Durability (Tính bền vững): Lưu trữ dữ liệu khi có sự cố về nguồn hoặc nút.

Trong cơ sở dữ liệu phân tán và sao chép, bản cập nhật giao tác cũng phải đảm bảo tính nhất quán của các giá trị trên tất cả các bản sao. Đối với một lớp dữ liệu khác, bạn có thể quyết định ưu tiên tính khả dụng hơn tính nhất quán.

Các doanh nghiệp dành nhiều nỗ lực trong việc thiết kế, phát triển và vận hành nhiều cơ sở dữ liệu được nhắm mục tiêu cho các nhu cầu dữ liệu đa dạng như vậy. May mắn thay, Aerospike xử lý truy cập dữ liệu đa dạng qua phổ tính khả dụng và nhất quán ở hiệu suất mili giây có thể dự đoán được, quy mô petabyte và khả năng phục hồi cao. Aerospike cũng loại bỏ nhu cầu về bộ nhớ đệm front-end thường cần thiết để đáp ứng các yêu cầu về độ trễ và thông lượng của các ứng dụng hiện đại vốn làm phức tạp hơn nữa việc thiết kế và triển khai.

# **Giao tác trong Aerospike**

Trong Aerospike, tất cả các thao tác trên bản ghi đơn lẻ đều có giao tác để đảm bảo. Mọi yêu cầu trên mỗi bản ghi, bao gồm cả những yêu cầu chứa nhiều thao tác trên một hoặc nhiều Thùng, thực hiện nguyên tố dưới một khóa bản ghi với mức cô lập và tính bền vững, và đảm bảo rằng tất cả các bản sao đều nhất quán.

Các giao tác không vượt qua nhiều ranh giới bản ghi. Một ứng dụng có thể loại bỏ hoặc giảm thiểu nhu cầu giao tác trên nhiều bản ghi (MRT – Multi-Record Transactions) bằng cách tận dụng mô hình dữ liệu phong phú của Aerospike bao gồm các loại Bản đồ và Danh sách lồng nhau cũng như các giao tác trên bản ghi đơn đa hoạt động của nó. Với mô hình dữ liệu thích hợp, có thể tránh được các giao tác trên nhiều bản ghi yêu cầu các phép kết (join) và các ràng buộc. Ví dụ: dữ liệu trong nhiều bảng trong cơ sở dữ liệu quan hệ có thể được mô hình hóa dưới dạng một bản ghi không chuẩn hóa duy nhất. Do đó, các hoạt động kiểm soát giao tác đa hoạt động như Begin, commit, Abort không phải là một phần của Aerospike API.

# **Cài đặt các giao tác đọc-ghi**

Aerospike cho phép nhiều thao tác đọc / ghi trên cùng một khóa trong một giao tác duy nhất. Tuy nhiên, ghi trong Aerospike là một thao tác đơn giản (set, add, append) và không thể là một chức năng tùy ý của một hoặc nhiều thùng trong bản ghi. Nói cách khác, logic cập nhật phức tạp không thể được gửi đến máy chủ để thực hiện trong một giao tác. Thiết kế Aerospike này giữ cho các thao tác đọc thông thường trở nên đơn giản và nhanh chóng có thể dự đoán được, đồng thời trì hoãn ứng dụng bất kỳ sự phức tạp nào của các giao tác đọc-ghi ít thường xuyên hơn.

Aerospike có hai tùy chọn cho các giao tác đọc-ghi chung:

* Sử dụng mẫu Record Read-Modify-Write (hoặc Check-and-Set) ở phía máy khách trong ứng dụng.
* Cài đặt update logic trong chức năng do người dùng xác định (UDF) nằm trên máy chủ. UDF, tương tự như một thủ tục được lưu trữ trong cơ sở dữ liệu, phải được cài đặt bởi quản trị viên và sau đó có thể được gọi thông qua API.

Các UDF, được triển khai trong Lua và nằm trên máy chủ, thực thi, phát triển, kiểm tra và thay đổi chậm hơn. Vì vậy, chúng không phù hợp nhất cho hầu hết các giao tác đọc-ghi trong các ứng dụng. Thay vì thực hiện ghi có điều kiện đối với siêu dữ liệu tạo của bản ghi, giao tác ghi có thể được thực hiện có điều kiện đối với giá trị đọc bằng Bộ lọc vị từ. Trong trường hợp đó, việc ghi sẽ chỉ thành công nếu giá trị đã đọc trước đó không thay đổi.

# **Giải quyết các giao tác nghi ngờ**

Ứng dụng phải biết rằng giao tác thành công hay thất bại để có thể thực hiện bất kỳ điều chỉnh cần thiết nào. Nếu giao tác không thành công, ứng dụng có thể thực hiện các bước thích hợp để thử lại giao tác (hoặc có logic cao hơn bao gồm can thiệp thủ công để thực hiện hành động thích hợp).

Trong những trường hợp hiếm hoi trong quá trình chuyển đổi cụm (khi cụm đang tách), không thể biết ngay giao dịch thành công hay thất bại. Trong trường hợp như vậy, giao dịch ghi hết thời gian với cờ “InDoubt” trong đối tượng ngoại lệ được đặt thành true, phản ánh kết quả vẫn chưa biết. Giao tác nghi ngờ có thể được giải quyết khi phân vùng dữ liệu bị ảnh hưởng hoạt động trở lại như một phần của quá trình khôi phục cụm tự động.

Kết quả của một giao dịch có thể không xác định được khi mạng đang trong giai đoạn phân tách, khi ghi nhận được trong cụm con thiểu số và cụm con khác không thể truy cập được trước khi tất cả các bản sao được biết là đã được cập nhật thành công. Kết quả chính xác của giao tác ghi không có sẵn ngay lập tức là thành công hay thất bại. Phân vùng bị ảnh hưởng cần phải hoạt động và có thể truy cập được sau khi khôi phục để giải quyết kết quả ghi.

Yêu cầu bị nghi ngờ sẽ hết thời gian. Không có một cách tích hợp nào để giải quyết một giao tác bị nghi ngờ và do đó, máy khách phải đưa ra kế hoạch của riêng mình để tìm ra kết quả của việc ghi. Một lần nữa, điều này phù hợp với nguyên lý thiết kế của Aerospike rằng để giữ cho các thao tác thông thường trở nên đơn giản và nhanh chóng có thể đoán trước được, nó sẽ trì hoãn bất kỳ sự phức tạp nào thêm vào các ứng dụng để xử lý các tình huống hiếm gặp.

Một cách điển hình để giải quyết thao tác ghi bị nghi ngờ là ghi lại ID giao tác (một máy khách được tạo id duy nhất trên toàn cục) trong bản ghi cùng với bản cập nhật thông qua một yêu cầu đa hoạt động. Thùng trong bản ghi giữ danh sách các ID của giao tác đã thành công, vì mỗi lần ghi sẽ thêm trước ID giao tác vào danh sách thông qua một yêu cầu đa thao tác nguyên tố. Ứng dụng có thể xác định giao tác ghi nghi ngờ có thành công hay không bằng cách kiểm tra xem ID giao tác có trong danh sách hay không. Danh sách được giữ cho không phát triển vô thời hạn bằng cách cắt nó theo một kích thước cụ thể. Kích thước cần được chọn thích hợp cho tần suất ghi của ứng dụng: nó phải đủ lớn để đảm bảo ID giao tác không bị cắt bớt khi thực hiện kiểm tra, nhưng không quá lớn vì nó sẽ làm tăng kích thước bản ghi và dung lượng đĩa. Danh sách có thể được xem như một giao tác trên bản ghi cụ thể đơn giản do ứng dụng quản lý.`

<https://aerospike.com/blog/developers-understanding-aerospike-transactions/>