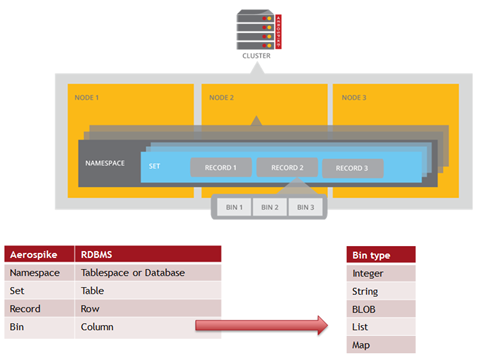
**Đặc điểm**

Aerospike là hệ quản trị lưu dữ liệu theo hàng, có nghĩa là mọi bản ghi (tương đương với một hàng trong cơ sở dữ liệu quan hệ) được xác định bằng duy nhất 1 khóa (key-value store). Khóa của bản ghi và các dữ liệu khác của nó sẽ nằm trong chỉ mục chính (primary index). Dữ liệu của bản ghi (record) sẽ nằm trong thiết bị lưu trữ đã được định nghĩa trước của không gian (namespace) mà nó chiếm giữ.

* Mỗi thùng chứa (bin) có thể giữ một kiểu dữ liệu dạng tập hợp (list, map, …) hoặc giữ một kiểu dữ liệu dạng đơn (int, double, string, …). Thùng chứa (bin) tương đương với một cột (thuộc tính) trong cơ sở dữ liệu quan hệ.
* Aerospike cho phép các truy vấn dựa trên giá trị bằng cách sử dụng các chỉ mục phụ (secondary index), trong đó các giá trị chuỗi và số nguyên được lập chỉ mục và tìm kiếm bằng cách lọc các giá trị bằng nhau (dành cho cả chuỗi và số nguyên) hoặc lọc trên phạm vi giá trị (chỉ dành cho số nguyên).
* Các chức năng do người dùng xác định (UDF) mở rộng chức năng và khả năng hoạt động của công cụ Cơ sở dữ liệu Aerospike.
* Trong Aerospike, khung kiến trúc Kết tập (aggregation) cho phép các hoạt động truy vấn linh hoạt và nhanh chóng. Giống như hệ thống MapReduce, các phép kết tập cho ra kết quả nhanh tương tự.
* Aerospike hỗ trợ lưu trữ, lập chỉ mục và truy vấn dữ liệu trong không gian dưới dạng GeoJSON.

# **Thao tác Key-value**

Aerospike cung cấp hỗ trợ đầy đủ cho mô hình lưu trữ tài liệu và mô hình lưu trữ khóa-giá trị. Trên thực tế, các hoạt động của nó cung cấp hỗ trợ rộng hơn nhiều so với các định nghĩa trong sách giáo khoa về các mô hình đó. Aerospike là một cơ sở dữ liệu hướng hàng (row-oriented), trong đó mọi bản ghi (record) được xác định duy nhất bằng một khóa. Khóa của bản ghi và siêu dữ liệu khác của nó nằm trong chỉ mục chính. Dữ liệu của bản ghi nằm trong thiết bị lưu trữ được xác định trước của Không gian tên mà nó chiếm.

## **Các mô hình được hỗ trợ:**

Aerospike hỗ trợ cả mô hình lưu trữ khóa-giá trị và lưu trữ tài liệu:

### **Mô hình lưu trữ khóa-giá trị (key-value store model)**

Các bản ghi được xác định duy nhất bởi một Khóa bao gồm tên của Bộ và khóa người dùng. Khóa người dùng là mã định danh chính cho bản ghi từ ứng dụng bạn đang xây dựng trên Aerospike. Mặc dù một giá trị trong mô hình lưu trữ khóa-giá trị là một khối dữ liệu không rõ ràng và không có hỗn hợp, tuy vậy thì các bản ghi bao gồm một hoặc nhiều thùng dữ liệu khối đã xác định hoặc chưa có kiểu dữ liệu.

### **Mô hình lưu trữ tài liệu (document store model)**

Một biến thể của mô hình lưu trữ khóa-giá trị, trong đó một bản ghi là một tài liệu dữ liệu duy nhất, được mã hóa bằng XML, YAML, JSON, BSON. Người dùng của Aerospike thường lưu trữ một tài liệu của dữ liệu trong loại dữ liệu Bản đồ (Map). Bản đồ Aerospike là một tập hợp JSON có thể chứa các kiểu dữ liệu khác nhau, bao gồm cả cấu trúc Bản đồ và Danh sách lồng vào nhau. Bản đồ sử dụng tính năng nén MessagePack một cách rõ ràng để lưu trữ hiệu quả. Trong khi mô hình lưu trữ tài liệu truyền thống lưu trữ một tài liệu dữ liệu trên mỗi bản ghi, bản ghi Aerospike có thể chứa nhiều ngăn và do đó có thể có nhiều kiểu dữ liệu cho mỗi bản ghi.

## **Các loại thao tác (Operation types)**

Aerospike cung cấp hai loại thao tác nguyên tố, hỗ trợ các trường hợp sử dụng rộng hơn so với hai mô hình được mô tả ở trên.

Bộ lọc thao tác: Thao tác có thể được áp dụng có điều kiện, dựa trên kết quả của một biểu thức bộ lọc. Nếu bộ lọc đánh giá là true, thao tác sẽ thực thi.

Vì Aerospike lưu trữ dữ liệu theo cặp khóa và giá trị, nên các thực thi sẽ được mô tả như là các thao tác Khóa-Giá trị:

### **Các thao tác trên bản ghi (Record Operations):**

các hàm tạo, đọc, cập nhật, xóa bản ghi (CRUD – create, read, update, delete).

* + Aerospike cung cấp các giao diện đơn giản để đọc dữ liệu và ghi dữ liệu vào một cụm (cluster). Chúng bao gồm các trường hợp sử dụng tiêu chuẩn CRUD.
  + Mọi máy khách (client) của Aerospike đều có thể thực hiện các thao tác ghi đầy đủ sau đây, bất kể nó được viết bằng ngôn ngữ nào:

|  |  |
| --- | --- |
| Thao tác (operation) | Chức năng |
| put | Viết một bản ghi. Thực hiện tương tự như create (tạo), update (cập nhật) hoặc replace (thay thế). |
| delete | Xóa bản ghi khỏi cơ sở dữ liệu. |
| get | Đọc bản ghi. Có thể chọn các thùng dữ liệu cụ thể. |
| add | Thêm (hoặc trừ) một giá trị trong thùng dữ liệu số nguyên. Được sử dụng để thực hiện bộ đếm, còn được gọi là increment. |
| append  prepend | Sửa đổi thùng dữ liệu kiểu chuỗi hoặc mảng byte. |
| touch | Tăng bộ đếm. |
| operate | Áp dụng nhiều thao tác thùng dữ liệu như một giao tác nguyên tố. |

### **Các thao tác trên giao tác (Transaction Operations):**

các thao tác dựa trên thùng được thực thi theo trình tự bằng cách sử dụng phương thức máy khách operation().

Bản ghi chứa một hoặc nhiều Thùng có nhiều kiểu dữ liệu khác nhau. Bản đồ và danh sách lồng nhau, còn được gọi là kiểu dữ liệu tập hợp (CDT – Collection data types) có thể được kết hợp để triển khai cấu trúc dữ liệu phức tạp dưới dạng tài liệu. Để cung cấp các tài liệu dữ liệu phức tạp này, mọi máy khách (client) của Aerospike cung cấp phương pháp sau:

Operate: Kết hợp nhiều thao tác (operation) trên một bản ghi (record) thành một giao tác (transaction). Các hoạt động bao gồm các hoạt động đọc và ghi trên Thùng, và bất kỳ hoạt động nào trên một kiểu dữ liệu, được lưu trữ trong một thùng.

Một trong những đối số của phương thức “OPERATE” là danh sách các thao tác trên Thùng có thứ tự để thực thi. Cú pháp được sử dụng bởi các máy khách Aerospike khác nhau tùy theo ngôn ngữ lập trình. Không giống như thao tác “GET” trên bản ghi sẽ trả về bản ghi, thao tác này trả về kết quả đầu ra từ mỗi thao tác con trên mỗi thùng. Điều này rất hữu ích để xác nhận quá trình xử lý giao tác và khắc phục sự cố logic dẫn đến giao tác thành công hoặc thất bại không mong muốn.

Mỗi lệnh gọi “OPERATE” dẫn đến một giao tác tuân thủ ACID bao gồm một hoặc nhiều thao tác trên Thùng có thứ tự trên bản ghi. Thao tác này là một giao tác nguyên tố đơn lẻ được xử lý bởi Aerospike dựa theo một khóa trên bản ghi với mức cô lập và tính bền vững. Để so sánh, mọi thao tác ghi dẫn đến một giao tác nguyên tố bao gồm một thao tác duy nhất.

Sau đây là danh sách đầy đủ các thao tác trên Thùng có thể là một phần của Giao tác có sử dụng thao tác. Bảng bên dưới chứa các thao tác hoặc liên kết đến danh sách đầy đủ các thao tác cho từng loại dữ liệu.

|  |  |  |
| --- | --- | --- |
| Kiểu dữ liệu  (data type) | Thao tác (Operation) | Chức năng |
| Tất cả | op\_put | Nâng cấp (tạo hoặc cập nhật) trên Thùng. Còn được gọi là write. |
| Tất cả | op\_get | Đọc trên Thùng. Còn được gọi là read. |
| Tất cả | op\_touch | Tăng bộ đếm trên bản ghi. |
| Tất cả | op\_delete | Xóa bản ghi khỏi cơ sở dữ liệu. |
| Số nguyên (Integer)  Số thực (Float) | op\_add | Cộng (hoặc trừ) một giá trị. Được sử dụng để thực hiện bộ đếm. Còn được gọi là increment. |
| Chuỗi (String) | op\_append  op\_prepend | Sửa đổi trên chuỗi. |
| Danh sách (List) | Danh sách các thao tác trên Lists | |
| Bản đồ (Map) | Danh sách các thao tác trên Maps | |
| Khối (Blob)/Bytes | Danh sách các thao tác trên Blob/Bytes | |
| HLL - HyperLogLog | Danh sách các thao tác trên HyperLogLog | |
| Geo | Danh sách các thao tác trên Geospatial | |

### **Thao tác đồng thời (Batch Operations):**

Yêu cầu hàng loạt gửi cùng một thao tác đọc cho nhiều bản ghi song song. Đây là một dạng truyền có thể sử dụng băng thông mạng hiệu quả hơn và giảm độ trễ khi một số lượng lớn các khóa chính tham gia vào.

Sử dụng các yêu cầu hàng loạt để:

* triển khai hiệu quả phép kết (join) phía máy khách.
* lưu trữ và sau đó truy xuất nhiều điểm dữ liệu trong một phép tính.
* xác định xem nhiều khóa có còn tồn tại hay không.

Aerospike hỗ trợ các yêu cầu hàng loạt này:

* get: Đọc nhiều bản ghi hoặc tùy chọn một thùng từ các bản ghi đó
* exists: Thực hiện kiểm tra siêu dữ liệu xem các khóa được chỉ định có tồn tại hay không.
* getHeader: Chỉ đọc siêu dữ liệu bản ghi. Không trả về Thùng dữ liệu.
* operate: Thực hiện một giao tác của các thao tác đọc đối với nhiều bản ghi.

## **Một số lưu ý trong việc triển khai**

Các đặc điểm của Aerospike cho thấy các phương pháp hay nhất để triển khai mô hình dữ liệu. Một số điểm cần ghi nhớ và lưu ý như:

* Aerospike là vô hình. Trước khi đưa dữ liệu vào cơ sở dữ liệu, bạn xác định không gian để liên kết dữ liệu với các loại phương tiện lưu trữ phần cứng. Bằng cách so sánh, các tập hợp (tương đương với mỗi bảng trong cơ sở dữ liệu quan hệ) và thùng (tương đương với mỗi cột trong cơ sở dữ liệu quan hệ) được tạo trong thời gian thực khi nó thực hiện các thao tác ghi vào Aerospike.
* Aerospike hỗ trợ các giao tác cho các bản ghi nguyên tố. Nhiều thao tác trên một hoặc nhiều thùng chứa bản ghi có thể được kết hợp thành một giao tác. Giao tác (transaction) được thực thi dưới sự kiểm soát của các khóa bản ghi với mức cô lập và tính bền vững.
* Mỗi loại dữ liệu của Aerospike đều có API riêng của các thao tác ghi và thao tác đọc phía máy chủ (server), chẳng hạn như hàm get\_by\_rank\_range () của loại dữ liệu Bản đồ.
* Đối với các thao tác phi nguyên tố, siêu dữ liệu tạo của bản ghi (số lượng sửa đổi dữ liệu) có thể được sử dụng để đảm bảo rằng dữ liệu đang được ghi không bị sửa đổi kể từ lần đọc cuối cùng. Sự tối ưu đồng thời này cho phép sử dụng mẫu kiểm tra và thiết lập (đọc-sửa đổi-ghi).

# **Các kiểu dữ liệu**

Các bản ghi gồm một hoặc nhiều Thùng. Mỗi thùng có thể chứa kiểu dữ liệu dạng đơn (số nguyên, chuỗi,…) hoặc dạng tập hợp (danh sách, bản đồ) hoặc dữ liệu dạng xác suất (HyperLogLog) hoặc dữ liệu không gian (GeoJSON)

## **Kiểu dữ liệu dạng đơn**

* Integer (số nguyên): giá trị số 64-bit. Mỗi số nguyên yêu cầu 8 bytes (64 bits) dung lượng lưu trữ. Phạm vi giá trị từ [-(263) đến 263-1]. Có thể áp dụng phép toán tăng increment để triển khai bộ đếm.
* Double (số thực): được lưu trữ ở định dạng IEEE-754 64-bit. Có thể áp dụng phép toán tăng increment.
* String (chuỗi): mặc định mã hóa dạng UTF-8. Kích thước bản ghi giới hạn 128kb
* Boolean: giữ giá trị true hoặc false, được lưu trữ dưới dạng một byte duy nhất trên máy chủ
* Blob/Bytes: Aerospike hỗ trợ hai loại dữ liệu Blob/Bytes chính. Có các thùng Blob dành cho mục đích chung và thùng Blob dành riêng cho ngôn ngữ.
  + Blob/Bytes chung: là mảng các byte có kích thước cụ thể. Mọi dữ liệu nhị phân thuộc bất kỳ loại nào đều có thể được lưu trữ. Các byte không được kết thúc bằng NULL. Có hỗ trợ các thao tác Bitwise.
  + Blob dành riêng cho ngôn ngữ (Language-Specific Serialized Blobs): chỉ có thể truy cập được từ thư viện ngôn ngữ của máy khách sử dụng. Chúng rất hữu ích để tuần tự hóa các đối tượng bằng ngôn ngữ đó vào cơ sở dữ liệu Aerospike. Hiện tại Aerospike hỗ trợ các loại Blob theo ngôn ngữ cụ thể sau: C#, Java, PHP, Python và Ruby.

## **Kiểu dữ liệu dạng tập hợp**

Các kiểu dữ liệu tập hợp (CDT) là các vùng chứa linh hoạt, không có giản đồ, có thể chứa dữ liệu dạng đơn hoặc lồng các tập hợp khác vào trong chúng. Các phần tử trong CDT có thể thuộc nhiều loại khác nhau. CDT là một tập hợp lớn của JSON, hỗ trợ nhiều kiểu dữ liệu hơn dưới dạng phần tử của tập hợp, chẳng hạn như số nguyên cho các khóa Bản đồ và dữ liệu nhị phân dưới dạng Danh sách.

* List (danh sách): là kiểu dữ liệu trừu tượng gồm 2 loại nhỏ là Danh sách có thứ tự và không có thứ tự. Cả hai loại đều chia sẻ cùng một API và có thể được truyền giữa nhau bằng thao tác set\_order(). Một danh sách có thể lưu trữ các kiểu dữ liệu dạng đơn, hoặc chứa các Bản đồ/Danh sách lồng nhau. Các thao tác trên Danh sách là cách tốt nhất để thao tác với danh sách trực tiếp trên cơ sở dữ liệu của Aerospike.
* Map (bản đồ): là kiểu dữ liệu trừu tượng gồm 3 loại nhỏ: Bản đồ có thứ tự Khóa (K-ordered), Bản đồ có thứ tự Khóa-Giá trị (KV-ordered) và Bản đồ không theo thứ tự. Tất cả các loại Bản đồ đều dùng chung một API và có thể truyền giữa nhau bằng thao tác set\_type(). Bản đồ có thể lưu trữ các kiểu dữ liệu dạng đơn hoặc chứa các Bản đồ/Danh sách lồng nhau. Thao tác trên bản đồ là tối ưu để thao tác với bản đồ trực tiếp trên cơ sở dữ liệu của Aerospike. Các phần tử trong Bản đồ được lưu trữ dựa trên kiểu sắp xếp. Các khóa bản đồ có thể thuộc loại Chuỗi, Byte, Số nguyên hoặc Số thực.

## **Chỉ mục không gian và truy vấn**

Sử dụng lưu trữ không gian, lập chỉ mục và truy vấn để cho phép truy vấn nhanh về các điểm trong một vùng, trên một vùng có chứa các điểm và các điểm trong bán kính. Định dạng GeoJSON để chỉ định và lưu trữ các đối tượng hình học GeoJSON, đồng thời đạt được tiêu chuẩn hóa dữ liệu và khả năng tương tác.

## **Dữ liệu xác suất**

Kiểu dữ liệu HyperLogLog cung cấp số lượng ước tính của các thành viên trong một tập dữ liệu lớn để ứng dụng của bạn tạo thành các ước tính nhanh, hợp lý của các thành viên trong liên hiệp hoặc giao nhau giữa nhiều thùng HyperLogLog. Các ước tính của HyperLogLog là sự cân bằng giữa độ chính xác hoàn toàn và khả năng tiết kiệm hiệu quả về dung lượng cũng như tốc độ xử lý các tập dữ liệu cực lớn. Dữ liệu HyperLogLog được trả về ứng dụng của bạn dựa trên một số ý tưởng cơ bản từ lý thuyết tập hợp.

# **Tính nhất quán (Consistency)**

Tính nhất quán mạnh mẽ đảm bảo rằng tất cả các lần ghi vào một bản ghi sẽ được áp dụng theo một thứ tự cụ thể (tuần tự) và các lần ghi sẽ không được sắp xếp lại hoặc bỏ qua, tức là chúng sẽ không bị mất. Trong tất cả các hình thức Nhất quán mạnh mẽ, việc ghi sẽ không bị mất (trong giới hạn của các lỗi phần cứng đồng thời).

Aerospike đảm bảo rằng dữ liệu sẽ không bị mất, với ba trường hợp ngoại lệ:

* Quá trình tạm dừng nút hơn 27 giây hoặc độ lệch đồng hồ giữa các cụm nút lớn hơn 27 giây.
* Máy chủ không đáng tin cậy đồng thời bị tắt khi "commit-to-device" không được bật.
* Nhiều lỗi lưu trữ phần cứng trước khi dữ liệu có thể tái tạo.

Trong mỗi trường hợp, Aerospike cố gắng cung cấp tính khả dụng tốt nhất và giảm thiểu các vấn đề có thể xảy ra. Trong trường hợp lệch đồng hồ, Aerospike liên tục theo dõi lượng sai lệch và sẽ cảnh báo nếu độ lệch trở nên lớn - và vô hiệu hóa rước khi xảy ra mất dữ liệu. Trong trường hợp máy chủ gặp sự cố hoặc tắt máy, Aerospike sẽ tự động nhanh chóng tái tạo và cân bằng lại dữ liệu trong trường hợp lỗi đầu tiên. Nếu sự cố xảy ra nhanh chóng, thì các phần của dữ liệu có thể bị đánh dấu là "chết" và cần sự can thiệp của người vận hành. Điều này là để cho phép sử dụng dữ liệu ở chế độ chỉ đọc hoặc cho phép tải lại các thay đổi gần đây.

# **Truy vấn (Query)**

Sử dụng các truy vấn nhanh đồng thời của Aerospike để thực hiện các tìm kiếm dựa trên giá trị bằng cách sử dụng các chỉ mục phụ. Các truy vấn dựa trên Chỉ mục phụ có thể trả về một tập hợp các bản ghi rất nhanh chóng, so với việc thay thế quét bảng.

Với chỉ mục phụ, có thể thực hiện truy vấn sau:

* Truy vấn ngang bằng với chỉ mục chuỗi hoặc số.
* Truy vấn phạm vi đối với các chỉ mục số. Tập hợp kết quả phạm vi là bao gồm (nghĩa là, cả hai giá trị đã chỉ định đều được bao gồm trong kết quả).
* Truy vấn điểm trong vùng (Point-In-Region) hoặc vùng chứa điểm (Region-Contain-Point) dựa trên chỉ mục địa lý.

Các kết quả của truy vấn chỉ mục phụ có thể tùy chọn được xử lý sau bằng Bộ lọc dự đoán của Aerospike hoặc Chức năng do người dùng xác định (UDF) của Aerospike trước khi trả về máy khách.

Sử dụng tính năng Bộ lọc dự đoán của Aerospike, bạn có thể:

* Lọc ra các bản ghi dựa trên dữ liệu meta bản ghi, chẳng hạn như thời gian cập nhật cuối cùng hoặc kích thước lưu trữ.
* Lọc ra các bản ghi dựa trên Thùng dữ liệu, chẳng hạn như số nguyên lớn hơn/nhỏ hơn trên các Thùng chuỗi.

Sử dụng các UDF của Aerospike, bạn có thể:

* Sử dụng bản đồ / giảm chức năng để thực hiện các phép kết hợp.
* Sử dụng các UDF trên bản ghi để tác động các bản ghi.
* Sử dụng toán tử không gian địa lý để trả về thông tin bản ghi vị trí địa lý

# **Phép tổng hợp (Aggregation)**

Sử dụng các UDF của Aerospike để tổng hợp các kết quả truy vấn theo kiểu phân tán. Khung tổng hợp Aerospike cho phép các hoạt động truy vấn linh hoạt và nhanh chóng. Khung chương trình này tương tự như một hệ thống MapReduce, trong đó một chức năng bản đồ ban đầu chạy trên một tập hợp và đưa ra kết quả. Kết quả đi qua một đường dẫn của các bước bản đồ tiếp theo hoặc các bước rút gọn và các bước tổng hợp.

Không giống như Hadoop hoặc các khung công tác khác sử dụng Java, tập hợp Aerospike được thực hiện bằng cách sử dụng Lua. Mỗi máy khách gửi một yêu cầu tổng hợp đến tất cả các máy chủ trong cụm, máy chủ này sẽ đếm kết quả một cách độc lập và trả về các kết quả riêng lẻ cho máy khách yêu cầu.

Khung tổng hợp Aerospike khác với các hệ thống khác vì Aerospike khuyến nghị chạy tập hợp dựa trên một chỉ mục; đây thực chất là một mệnh đề "WHERE". Lọc theo chỉ mục duy trì hiệu suất cao. Aerospike hỗ trợ tổng hợp các bảng và toàn bộ không gian tên. Sau đó, khách hàng chạy một giai đoạn giảm cuối cùng, cũng trong Lua, để tổng hợp kết quả.

Trường hợp sử dụng:

* Triển khai các hàm tổng hợp như SUM, COUNT, MIN, MAX dưới dạng các UDF do người dùng xác định.
* Bảng điều khiển thời gian thực.
* Sử dụng các chỉ mục phụ trên Thùng với thời gian cập nhật, tổng hợp nhanh chóng thu thập số liệu thống kê về các bản ghi được thay đổi gần đây.

Ảnh có chứa văn bản

Mô tả được tạo tự độngĐể triển khai phép tổng hợp:

* Tạo một truy vấn.
  + Tạo chỉ mục trên thùng.
  + Chèn bản ghi vào thùng được lập chỉ mục.
* Tạo một mô-đun tổng hợp trong Lua.
* Đặt đường dẫn mô-đun tổng hợp.
* Đăng ký mô-đun với cụm Aerospike.
* Tạo một truy vấn tổng hợp với một vị từ (mệnh đề where).