Báo cáo

Aerospike

Aerospike là một cơ sở dữ liệu NoSQL phân tán, có thể mở rộng.:

- Tạo ra một nền tảng linh hoạt, có thể mở rộng cho các ứng dụng web.
- Cung cấp các tính năng mạnh mẽ và độ tin cậy (như trong ACID) mong đợi từ các cơ sở dữ liệu truyền thống.
- Cung cấp hiệu quả hoạt động.

Kiến trúc Aerospike bao gồm ba lớp:

- **Client Layer**: bao gồm các thư viện client mã nguồn mở, thực hiện các API của Aerospike, theo dõi các nút và biết dữ liệu nằm trong cụm nào.
- **Clustering và Data Distribution Layer**: quản lý truyền thông cụm và tự động hóa quá trình nhân bản, sao chép, đồng bộ hóa data center, cân bằng lại dữ liệu và di chuyển dữ liệu.
- **Data Storage Layer**: lưu trữ dữ liệu trong DRAM và Flash để phục hồi nhanh chóng

Data model

Namespaces

Không gian tên là các vùng chứa dữ liệu cấp cao nhất. Một không gian tên thực sự có thể là một phần của một cơ sở dữ liệu hoặc một nhóm cơ sở dữ liệu. Cách bạn thu thập dữ liệu trong không gian tên liên quan đến cách dữ liệu được lưu trữ và quản lý. Một không gian tên chứa các bản ghi, chỉ mục, và các chính sách. Chính sách chỉ định hành vi không gian tên, bao gồm:

- Cách dữ liệu được lưu trữ: trên DRAM hoặc đĩa
- Bao nhiêu bản sao tồn tại cho một bản ghi.
- Khi bản ghi hết hạn.

Sets

Trong không gian tên, các bản ghi thuộc về các bộ chứa logic được gọi là sets. Sets cho phép các ứng dụng hợp lý hóa các bản ghi trong collection. Sets kế thừa các chính sách được xác định bởi không gian tên của chúng và có thể xác định các chính sách bổ sung hoặc các hoạt động cụ thể cho tập hợp. Ví dụ, chỉ số thứ cấp chỉ có thể được chỉ định trên dữ liệu cho một tập hợp cụ thể, hoặc một hoạt động quét có thể được thực hiện trên một tập cụ thể.

Records

Cơ sở dữ liệu Aerospike là lưu trữ một row với tập trung vào các bản ghi cá nhân (các hàng RDBMS). Một bản ghi là đơn vị cơ bản của lưu trữ trong cơ sở dữ liệu. Các bản ghi có thể thuộc về một không gian tên hoặc một tập trong không gian tên. Các bản ghi sử dụng một chìa khóa là mã nhận diện duy nhất của chúng.

Records bao gồm:

- **key** : Định danh duy nhất. Các bản ghi có thể được định vị bằng cách sử dụng một băm của khóa, được gọi là tiêu hóa.
- **Metadata**: Ghi lại thông tin về phiên bản và thời hạn kết nối được gọi là thời gian sống.
- **Bins**: tương đương với các trường trong RDBMS.

Storing Data

Aerospike hỗ trợ các loại dữ liệu sau:

- integers
- doubles
- strings
- blobs
- lists
- maps
- geoJSON
- natively serialized types

Install Aerospike

install aerosspike theo hướng dẫn trong link: http://www.aerospike.com/docs/operations/install/linux/ubuntu

ElasticSearch

Elasticsearch là một nền tảng tìm kiếm gần thời gian thực. Điều này có nghĩa là có độ trễ nhẹ (thường là một giây) kể từ khi bạn lập chỉ mục tài liệu cho đến khi nó có thể tìm kiếm được.

Trong ES, tất cả các document được hiển thị trong JSON format. Nó được xây dựng trên Lucene – phần mềm tìm kiếm và trả về thông tin (information retrieval software) với hơn 15 năm kinh nghiệm về full text indexing and searching.

ES thực sự đặc biệt chính là nhờ vào khả năng phục hồi thông tin của nó. Sự kết hợp

của storage và querying/aggregation service đã làm cho ES thực sự đặc biệt và khác xa 1 công cụ chỉ lưu trữ văn bản.

Với bản chất của nó, một điều quan trọng cần biết đó là: khi nào thì nên sử dụng ElasticSearch?. Một số trường hợp nên sử dụng ES:

- Tìm kiếm text thông thường - Searching for pure text (textual search)
- Tìm kiếm text và dữ liệu có cấu trúc - Searching text and structured data (product search by name + properties)
- Tổng hợp dữ liệu - Data aggregation
- Tìm kiếm theo tọa độ - Geo Search
- Lưu trữ dữ liệu theo dạng JSON - JSON document storage

Một số khái niệm

Cluster

Một cụm là tập hợp của một hoặc nhiều nút (máy chủ) chứa cùng toàn bộ dữ liệu của bạn và cung cấp các khả năng lập chỉ mục và tìm kiếm liên kết trên tất cả các nút. Một cụm được xác định bởi một tên duy nhất mà theo mặc định là "elasticsearch". Tên này rất quan trọng vì một node chỉ có thể là một phần của cluster nếu node được thiết lập để join cluster với tên của nó.

Node

Nút là một máy chủ nằm trong cụm của bạn, lưu trữ dữ liệu của bạn và tham gia vào các khả năng lập chỉ mục và tìm kiếm của cụm. Giống như một cụm, một nút được xác định bởi một tên mà theo mặc định là ngẫu nhiên Universally Unique IDentifier (UUID) được gán cho nút khi khởi động. Bạn có thể định nghĩa bất kỳ tên nút nào bạn muốn nếu bạn không muốn mặc định. Tên này rất quan trọng cho mục đích quản trị mà bạn muốn xác định máy chủ nào trong mạng của mình tương ứng với các nút nào trong cụm Elasticsearch của bạn.

Một node có thể được cấu hình để join một cluster cụ thể theo tên cluster. Theo mặc định, mỗi nút được thiết lập để kết nối cụm sao có tên elasticsearch nghĩa là nếu bạn khởi động một số nút trên mạng của mình và giả sử họ có thể khám phá nhau - tất cả sẽ tự động tạo và tham gia một cụm duy nhất có tên elasticsearch.

Index

Một chỉ mục là tập hợp các tài liệu có đặc điểm tương tự nhau. Ví dụ: bạn có thể có một chỉ mục cho dữ liệu khách hàng, một chỉ mục khác cho danh mục sản phẩm, và một chỉ mục khác cho dữ liệu đặt hàng. Một chỉ mục được xác định bởi tên (phải là tất cả chữ thường) và tên này được sử dụng để chỉ vào chỉ mục khi thực hiện các thao tác lập chỉ mục, tìm kiếm, cập nhật và xóa đối với các tài liệu trong đó.

Trong một cluster đơn, bạn có thể định nghĩa nhiều chỉ mục mà bạn muốn.

Type

Trong một chỉ mục, bạn có thể định nghĩa một hoặc nhiều loại. Có thể là logiccal/partition của chỉ mục của bạn có nghĩa là hoàn toàn tùy thuộc vào bạn. Nói chung, một loại được định nghĩa cho các tài liệu có một tập hợp các trường phổ biến.

Document

Document là một đơn vị cơ bản của thông tin có thể được lập chỉ mục. Document này được biểu diễn bằng định dạng JSON (JavaScript Object Notation).

Trong một index / type, bạn có thể lưu trữ nhiều documents như bạn muốn. Lưu ý rằng mặc dù các documents nằm trên một chỉ mục, một tài liệu thực sự phải được lập index / assigned cho một loại bên trong một chỉ mục.

Shards

Tập con các documents của 1 Index. Một Index có thể được chia thành nhiều shard.

Ưu và nhược điểm của ElasticSearch

Ưu điểm

Điều đầu tiên là tốc độ. ES có perfomance rất tốt. Nó được xây dựng trên Lucene và khả năng mở rộng truy vấn song song bên trong một cluster rất tốt (spanning queries in parallel inside a cluster).

Một ưu điểm khác của ElasticSearch là có thể sắp xếp kết quả truy vấn theo Relevance (sự liên quan). Theo mặc định, ES sử dụng thuật toán TF/IDF tương tự để tính toán relevance. Nếu bạn không biết relevance là gì, hãy xem trang này trong tài liệu ES.

Cuối cùng, ES có thể rất hữu ích để tạo ra số liệu thống kê tổng hợp (aggregate statistics), và với chút ít nỗ lực, Search API có thể linh hoạt đáp ứng yêu cầu của bạn.

Nhược điểm

ElasticSearch rất tốt trong việc tìm kiếm và tổng hợp data, nhưng nếu bạn đang sở hữu môi trường thường xuyên ghi dữ liệu (writing operations environment), ES có thể sẽ không phải lựa chọn tốt nhất của bạn.

Indexing

ES còn có những điểm khác giúp searching document hiệu quả hơn. Khi lưu trữ document trong ES, nó tạo ra một số internal data structures làm cho query perfom tốt hơn. Tôi sẽ nói về một số chi tiết cơ bản về ES indexing technique.

Mỗi document gửi tới ES được lưu trữ qua một thuật toán và sau đó được gửi đến shard. ES cố gắng để phân tán document thông qua các shard.

Khi lưu trữ document, ES tạo ra inverted index, map các thuật ngữ/từ khóa xuất hiện trong document này tới chính document đó.

Khi sử dụng inverted index, nó có thể tìm kiếm thông qua terms như một binary tree (sử dụng thứ tự chữ cái) làm giảm thời gian tìm kiếm.

Một điều quan trọng khi lưu trữ document đó là được quyết định cách tốt nhất để lưu trữ chúng giúp nâng cao tốc độ truy vấn. Khi thiết kế các giải pháp sử dụng Elastic-Search, điều đáng lưu tâm nhất khi lưu trữ document chính là: tôi sẽ truy vấn document này như thế nào? "First query" này tiếp cận với việc sử dụng cho tất cả các khả năng ES indexing để thực hiện truy vấn cực kỳ nhanh chóng.

Querying

Một tính năng mạnh khác của ElasticSearch đó chính là cung cấp tất cả query type. Có gần 40 query type và có lẽ là một trong những loại này sẽ đáp ứng hoàn hảo hầu hết các nhu cầu.

ES có phrase queries for textual search, geo queries based on coordinates, numeric range queries. Chúng có thể sẽ rất hữu ích cho các dữ liệu tổng hợp và nhiều hơn nữa.

Redis

Redis là một nguồn mở,lưu trữ key-value và giải pháp xây dựng hiệu năng cáo để xây dựng các ứng dụng web có khả năng mở rộng cao

Redis có ba đặc điểm chính:

- Redis giữ cơ sở dữ liệu của nó hoàn toàn trong bộ nhớ, sử dụng đĩa chỉ cho sự kiên trì.
- Redis có một tập các loại dữ liệu tương đối phong phú khi so sánh với nhiều kiểu lưu trữ key-value.
- Redis có thể sao chép dữ liệu tới bất kỳ số máy slaves nào.

Ưu điểm

- **Exceptionally fast**: Redis rất nhanh và có thể thực hiện khoảng 110000 sets mỗi giây, khoảng 81000 gets mỗi giây.
- **Supports rich data types**: hỗ trợ hầu hết các kiểu dữ liệu mà các nhà phát triển đã biết như list, set, sort set, và hash. Điều này làm cho nó dễ dàng để giải quyết một loạt các vấn đề như chúng ta biết mà vấn đề có thể được xử lý tốt hơn bằng cách loại dữ liệu.
- **Operations are atomic**: Tất cả các hoạt động Redis là nguyên tử, trong đó đảm bảo rằng nếu hai khách hàng đồng thời truy cập, Redis server sẽ nhận được giá trị cập nhật.

- **Multi-utility tool**: Redis là một công cụ đa tiện ích và có thể được sử dụng trong một số trường hợp sử dụng như bộ nhớ đệm, tin nhắn hàng đợi (Redis natively hỗ trợ Xuất bản / Theo dõi), bất kỳ dữ liệu ngắn ngủi trong ứng dụng của bạn, chẳng hạn như web phiên ứng dụng, trang web hit đếm ....

Redis - DataTypes

Redis hỗ trợ 5 kiểu dữ liệu chính:

- Strings
  Redis string là một chuỗi các byte. Strings trong Redis được an toàn nhị phân, nghĩa là chúng có một chiều dài được biết đến không xác định bằng bất kỳ ký tự chấm dứt đặc biệt. Vì vậy, bạn có thể lưu trữ bất cứ thứ gì lên đến 512 MB trong một chuỗi.

- Hashes
  Hashes Redis là một tập hợp các cặp giá trị quan trọng. Redis Hashes là ánh xạ giữa các string và giá trị string. Do đó, chúng được sử dụng để đại diện cho các đối tượng.

- Lists
  List trong Redis chỉ đơn giản là danh sách các chuỗi, được sắp xếp theo thứ tự chèn. Bạn có thể thêm các phần tử vào một danh sách Redis có thể thêm ở trên đầu hoặc cuối danh sách.

- Sets
  Set trong Redis là một tập có thứ tự các chuỗi. Trong Redis, bạn có thể thêm, xóa, và kiểm tra sự tồn tại của các phần tử trong thời gian O(1).

- Sorted Sets
  Redis Sorted Sets cũng tương tự như Redis Set, các tập không lặp lại của Strings. Sự khác biệt là, mỗi phần tử của một Sorted Set gắn liền với điểm số, được sử dụng để lấy các thiết lập được sắp xếp có trật tự, từ nhỏ nhất đến số điểm cao nhất. Trong khi các thành viên là duy nhất, điểm số có thể được lặp đi lặp lại.

Install Redis

cài đặt theo hướng dẫn trong link: https://www.tutorialspoint.com/redis/redis_environment.htm