Learning Ceph

Reliable Autonomic Distributed Object Store (RADOS) là nền tảng của cụm lưu trữ Ceph. Mọi thông tin trong Ceph được lưu trữ dưới dạng các Object và lớp RADOS có trách nhiệm lưu trữ những Object này, bất kể loại dữ liệu của chúng. Lớp RADOS đảm bảo dữ liệu luôn ở trạng thái nhất quán và đáng tin cậy. Để đảm bảo tính nhất quán của dữ liệu, RADOS thực hiện sao chép dữ liệu, phát hiện và phục hồi lỗi, di chuyển dữ liệu và cân bằng lại giữa các node trong cụm.

Ngay khi ứng dụng thực hiện một thao tác ghi vào cụm Ceph, dữ liệu được lưu trữ trong thiết bị lưu trữ Object Ceph (Ceph Object Storage Device - OSD) dưới dạng các Object. Đây là thành phần duy nhất trong cụm Ceph nơi lưu trữ thực sự của dữ liệu người dùng và truy vấn dữ liệu khi có nhu cầu đọc ghi. Thông thường, một tiến trình OSD được liên kết với một đĩa vật lý trong cụm. Do đó, tổng số đĩa vật lý trong cụm Ceph thường là số tiến trình OSD đang hoạt động để lưu trữ dữ liệu người dùng vào mỗi đĩa vật lý.

Ceph monitors – MONs theo dõi tình trạng của toàn bộ cụm bằng cách duy trì một bản đồ trạng thái của cụm, bao gồm các bản đồ OSD, MON, PG và CRUSH. Tất cả các node trong cụm báo cáo cho Mon và chia sẻ thông tin về mọi thay đổi trong trạng thái của chúng. Mon duy trì một bản đồ thông tin riêng cho mỗi thành phần. Mon không lưu trữ dữ liệu thực tế, điều này là công việc của OSD.

Thư viện librados cung cấp một cách tiện lợi để truy cập vào RADOS với sự hỗ trợ của các ngôn ngữ lập trình PHP, Ruby, Java, Python, C và C++. Nó cung cấp một giao diện native cho cụm lưu trữ Ceph, RADOS và là cơ sở cho các dịch vụ khác như RBD, RGW, cũng như giao diện POSIX cho CephFS. API librados hỗ trợ truy cập trực tiếp vào RADOS và cho phép tạo giao diện riêng của mình cho cụm lưu trữ Ceph.

Ceph Block Device, được biết đến với tên gọi RADOS block device (RBD), cung cấp lưu trữ khối, có thể được ánh xạ, định dạng và gắn kết giống như bất kỳ đĩa khác trên máy chủ. Một thiết bị khối Ceph được trang bị các tính năng lưu trữ doanh nghiệp như Thin provisioning. Đây là một tính năng quan trọng trong lưu trữ doanh nghiệp. Khi sử dụng Thin provisioning, không cần phải cung cấp toàn bộ dung lượng lưu trữ cho thiết bị từ đầu. Thay vào đó, chỉ cung cấp một phần dung lượng ban đầu, và dung lượng thực sự được cấp phát dựa trên nhu cầu sử dụng thực tế của ứng dụng. Điều này giúp tối ưu hóa sử dụng không gian lưu trữ và giảm thiểu lãng phí.

Ceph Object Gateway, còn được gọi là RADOS gateway (RGW), cung cấp một giao diện API RESTful, tương thích với Amazon S3 (Simple Storage Service) và OpenStack Object Storage API (Swift). RGW cReliable Autonomic Distributed Object Store (RADOS) là nền tảng cơ sở của cụm lưu trữ Ceph. Trong Ceph, mọi dữ liệu được lưu trữ dưới dạng các Object và lớp RADOS có trách nhiệm lưu trữ những Object này, bất kể loại dữ liệu của chúng. Lớp RADOS đảm bảo tính nhất quán và đáng tin cậy của dữ liệu. Để đảm bảo tính nhất quán của dữ liệu, RADOS thực hiện sao chép dữ liệu, phát hiện và khắc phục lỗi, di chuyển dữ liệu và cân bằng lại trên các node trong cụm.

Ngay khi ứng dụng thực hiện một thao tác ghi vào cụm Ceph, dữ liệu được lưu trữ trong Ceph Object Storage Device (OSD) dưới dạng các Object. OSD là thành phần duy nhất trong cụm Ceph nơi lưu trữ thực sự của dữ liệu người dùng và cung cấp dữ liệu tương tự khi một khách hàng yêu cầu thao tác đọc. Thông thường, mỗi tiến trình OSD được liên kết với một đĩa vật lý trong cụm.

Ceph monitors (MONs) theo dõi tình trạng của toàn bộ cụm bằng cách duy trì một bản đồ trạng thái của cụm, bao gồm các bản đồ OSD, MON, PG và CRUSH. Tất cả các node trong cụm báo cáo cho các node theo dõi và chia sẻ thông tin về mọi thay đổi trong trạng thái của chúng. Mỗi node theo dõi duy trì một bản đồ thông tin riêng cho từng thành phần. Các node theo dõi không lưu trữ dữ liệu thực tế, điều này là công việc của OSD.

Thư viện librados cung cấp một cách tiện lợi để truy cập vào RADOS với sự hỗ trợ của các ngôn ngữ lập trình PHP, Ruby, Java, Python, C và C++. Nó cung cấp một giao diện native cho cụm lưu trữ Ceph, RADOS và là cơ sở cho các dịch vụ khác như RBD, RGW, cũng như giao diện POSIX cho CephFS. API của librados hỗ trợ truy cập trực tiếp vào RADOS và cho phép tạo giao diện riêng của mình cho cụm lưu trữ Ceph.

Ceph Block Device, trước đây được biết đến với tên RADOS block device (RBD), cung cấp lưu trữ khối, có thể được ánh xạ, định dạng và gắn kết giống như bất kỳ đĩa khác trên máy chủ. Một thiết bị khối Ceph được trang bị các tính năng lưu trữ doanh nghiệp như Thin provisioning. Đây là một tính năng quan trọng trong lưu trữ doanh nghiệp. Khi sử dụng Thin provisioning, không cần phải cung cấp toàn bộ dung lượng lưu trữ cho thiết bị từ đầu. Thay vào đó, chỉ cung cấp một phần dung lượng ban đầu, và dung lượng thực sự được cấp phát dựa trên nhu cầu sử dụng thực tế của ứng dụng. Điều này giúp tối ưu hóa sử dụng không gian lưu trữ và giảm thiểu lãng phí.

Ceph Object Gateway, còn được gọi là RADOS gateway (RGW), cung cấp một giao diện RESTful API, tương thích với Amazon S3 (Simple Storage Service) và OpenStack Object Storage API (Swift). RGW cũng hỗ trợ các dịch vụ đa người dùng và xác thực từ khóa OpenStack Keystone.

Ceph File System - CephFS cung cấp một hệ thống tệp phân tán có độ lớn bất kỳ . CephFS dựa trên Ceph MDS để theo dõi cấu trúc thư mục của tệp.

1. RADOS

Ceph RADOS (Reliable Autonomic Distributed Object Store) là trái tim của hệ thống lưu trữ Ceph. RADOS cung cấp tất cả các tính năng quan trọng cho Ceph, bao gồm hệ thống lưu trữ object phân tán, khả năng cao, đáng tin cậy, no single point of failure, tự phục hồi và tự quản lý, và nhiều tính năng khác. Do đó, lớp RADOS có ý nghĩa đặc biệt trong kiến trúc lưu trữ Ceph. Các phương pháp truy cập dữ liệu của Ceph, chẳng hạn như RBD, CephFS, RADOSGW và librados, đều hoạt động trên lớp RADOS.

Khi cụm Ceph nhận yêu cầu ghi dữ liệu của người dùng, thuật toán CRUSH tính toán vị trí và quyết định nơi dữ liệu nên được ghi. Thông tin này sau đó được chuyển đến lớp RADOS để xử lý tiếp. Dựa trên tập luật CRUSH, RADOS phân phối dữ liệu đến tất cả các nút trong cụm dưới dạng các object nhỏ. Cuối cùng, những object này được lưu trữ trên các OSD.

Khi được cấu hình với một hệ số sao chép lớn hơn một, RADOS đảm bảo tính tin cậy của dữ liệu. Đồng thời, nó sao chép object, tạo bản sao và lưu trữ chúng trong một khu vực lỗi khác nhau, tức là các bản sao của cùng một object không nên nằm trên cùng một khu vực lỗi. Tuy nhiên, để tùy chỉnh và đảm bảo tính tin cậy cao hơn, nên điều chỉnh tập luật CRUSH phù hợp với nhu cầu và yêu cầu cơ sở hạ tầng.

Ngoài việc lưu trữ và sao chép object trên cụm, RADOS cũng đảm bảo trạng thái đồng nhất của object. Trong trường hợp không nhất quán của object, quá trình khôi phục được thực hiện với các bản sao object còn lại. Hoạt động này được thực hiện tự động và không nhìn thấy bởi người dùng, do đó cung cấp khả năng tự quản lý và tự phục hồi cho Ceph.

\* RADOS lưu trữ dữ liệu dưới dạng các object trong một pool

RADOS lưu trữ dữ liệu dưới dạng các object trong một pool. Một pool trong RADOS là một không gian lưu trữ ảo được sử dụng để nhóm các object có liên quan với nhau. Nó cung cấp một cơ chế để tổ chức và quản lý dữ liệu trong Ceph cluster.

Mỗi pool được đặt tên và có các thuộc tính riêng, bao gồm các thiết lập về sao chép, quyền truy cập và cấu hình bổ sung. Các object trong cùng một pool thường chia sẻ các thiết lập sao chép và quyền truy cập chung.

Thông qua việc sử dụng pool, có thể tạo ra các không gian lưu trữ riêng biệt để phân loại và quản lý dữ liệu theo nhu cầu của . có thể tạo ra các pool để lưu trữ các loại dữ liệu khác nhau như ảnh, video, tệp tin hệ thống, và nhiều hơn nữa.

Việc sử dụng pool trong RADOS giúp tăng tính linh hoạt và hiệu quả của việc quản lý dữ liệu trong Ceph cluster, cho phép tổ chức và quản lý dữ liệu một cách tối ưu và dễ dàng.

1. Ceph Object Storage Device

Ceph Object Storage Device (OSD) là một trong những khối xây dựng quan trọng nhất của cụm lưu trữ Ceph. Nó lưu trữ dữ liệu thực tế trên ổ đĩa vật lý của mỗi nút trong cụm dưới dạng các object. Đa số công việc trong một cụm Ceph được thực hiện bởi các tiến trình Ceph OSD. Đây là những thành phần chính thực sự lưu trữ dữ liệu người dùng. Bây giờ chúng ta sẽ thảo luận về vai trò và trách nhiệm của một tiến trình Ceph OSD.

Ceph OSD lưu trữ tất cả dữ liệu khách hàng dưới dạng các object và phục vụ dữ liệu đó cho khách hàng khi họ yêu cầu. Một cụm Ceph bao gồm nhiều OSD. Đối với bất kỳ hoạt động đọc hoặc ghi nào, khách hàng yêu cầu các bản đồ cụm từ các monitor, sau đó họ có thể tương tác trực tiếp với OSD để thực hiện các hoạt động nhập/xuất, mà không cần sự can thiệp của một monitor. Điều này làm cho quá trình giao dịch dữ liệu nhanh chóng, vì khách hàng tạo ra dữ liệu có thể ghi trực tiếp vào OSD lưu trữ dữ liệu mà không cần bất kỳ lớp xử lý dữ liệu bổ sung nào. Cơ chế lưu trữ và truy xuất dữ liệu này tương đối độc đáo trong Ceph so với các giải pháp lưu trữ khác.

Các tính năng cốt lõi của Ceph, bao gồm độ tin cậy, cân bằng, khôi phục và đồng nhất, đều đi kèm với OSD. Dựa trên kích thước sao chép được cấu hình, Ceph đảm bảo tính tin cậy bằng cách sao chép mỗi object nhiều lần trên các nút trong cụm, làm cho chúng có tính sẵn có cao và kháng lỗi. Mỗi object trong OSD có một bản sao chính và một số bản sao phụ, được phân tán trên tất cả các OSD khác. Vì Ceph là một hệ thống phân tán và các object được phân tán trên nhiều OSD, mỗi OSD đóng vai trò của OSD chính cho một số object và đồng thời trở thành OSD phụ cho các object khác. OSD phụ vẫn nằm dưới sự kiểm soát của OSD chính; tuy nhiên, chúng có khả năng trở thành OSD chính. Bắt đầu từ phiên bản Ceph Firefly (0.80), đã được thêm một cơ chế bảo vệ dữ liệu mới được gọi là mã hóa.

Trong trường hợp xảy ra sự cố với đĩa, tiến trình Ceph OSD thông minh kết nối với các OSD khác để thực hiện các hoạt động khôi phục. Trong quá trình này, OSD phụ nắm giữ các bản sao của object bị lỗi sẽ được thăng cấp thành OSD chính, đồng thời, các bản sao mới của object được tạo ra trong quá trình khôi phục OSD, mà hoàn toàn không ảnh hưởng đến khách hàng. Điều này làm cho cụm Ceph trở nên đáng tin cậy và nhất quán. Một triển khai cụm Ceph tiêu chuẩn tạo ra một tiến trình OSD cho mỗi ổ đĩa vật lý trên mỗi nút trong cụm, đó là một thực práctice khuyến nghị. Tuy nhiên, OSD hỗ trợ việc triển khai linh hoạt một tiến trình OSD cho mỗi ổ đĩa, mỗi máy chủ hoặc mỗi RAID. Đa số các triển khai cụm Ceph trong môi trường JBOD sử dụng một tiến trình OSD cho mỗi ổ đĩa vật lý.

1. Ceph monitors

Các monitor Ceph

Như tên gọi, monitor Ceph chịu trách nhiệm giám sát sức khỏe của toàn bộ cụm. Đây là các tiến trình duy trì trạng thái thành viên của cụm bằng cách lưu trữ thông tin quan trọng về cụm, trạng thái các nút đồng nghiệp và thông tin cấu hình cụm. Monitor Ceph thực hiện nhiệm vụ của mình bằng cách duy trì một bản sao chính của cụm. Bản đồ cụm bao gồm các bản đồ monitor, OSD, PG, CRUSH và MDS. Tất cả các bản đồ này được gọi chung là bản đồ cụm. Hãy xem qua một cách ngắn gọn về chức năng của mỗi bản đồ:

1. Bản đồ monitor (Monitor map): Chứa thông tin về các monitor trong cụm và trạng thái của chúng.

2. Bản đồ OSD (OSD map): Được sử dụng để theo dõi trạng thái và sẵn sàng của các OSD trong cụm.

3. Bản đồ PG (Placement Group map): Quản lý việc phân bố các object trong các nhóm đặt (PG) để đảm bảo cân bằng tải và hiệu suất.

4. Bản đồ CRUSH: Được sử dụng để ánh xạ object và OSD vào các nút vật lý trong cụm, đảm bảo phân tán và cân bằng tải.

5. Bản đồ MDS (Metadata Server map): Sử dụng cho Ceph File System (CephFS) để theo dõi trạng thái và sẵn sàng của các máy chủ dữ liệu (MDS).

Bằng cách duy trì và cập nhật các bản đồ này, monitor Ceph đảm bảo sự nhất quán và sẵn sàng của cụm, giúp các thành viên trong cụm làm việc cùng nhau một cách hiệu quả.

Ceph monitor không lưu trữ và cung cấp dữ liệu cho khách hàng, thay vào đó, nó cung cấp các bản đồ cụm cập nhật cho khách hàng cũng như các nút khác trong cụm. Khách hàng và các nút khác trong cụm định kỳ kiểm tra với các monitor để lấy bản sao mới nhất của các bản đồ cụm.

Monitor là các tiến trình nhẹ nhàng và thường không đòi hỏi một lượng lớn tài nguyên hệ thống. Một máy chủ mức giá thấp, mức đầu với một lượng CPU, bộ nhớ và Ethernet gigabit đủ cho hầu hết các tình huống. Một nút monitor nên có đủ không gian đĩa để lưu trữ các nhật ký cụm, bao gồm nhật ký OSD, MDS và monitor.

Một cụm khỏe mạnh thông thường tạo ra các nhật ký từ vài MB đến vài GB; tuy nhiên, yêu cầu lưu trữ cho các nhật ký tăng lên khi mức độ chi tiết/gỡ lỗi được tăng lên cho một cụm. Có thể cần một số GB không gian đĩa để lưu trữ nhật ký.

*Lưu ý: quan trọng là đảm bảo rằng ổ đĩa hệ thống không được đầy, nếu không cụm có thể gặp vấn đề. Chính sách quay vòng nhật ký theo lịch trình cùng việc theo dõi sử dụng hệ thống tệp thường xuyên được khuyến nghị, đặc biệt là đối với các nút chứa monitor vì việc tăng mức độ chi tiết/gỡ lỗi có thể dẫn đến việc tạo ra nhật ký lớn với tốc độ trung bình 1 GB mỗi giờ.*

Một cụm Ceph điển hình bao gồm nhiều hơn một nút monitor. Kiến trúc Ceph với nhiều monitor phát triển quorum và cung cấp sự nhất quán cho quyết định phân tán trong các cụm bằng cách sử dụng thuật toán Paxos. Số lượng monitor trong cụm của nên là một số lẻ; yêu cầu tối thiểu là một nút monitor và số lượng được khuyến nghị là ba. Vì monitor hoạt động trong quorum, luôn phải có hơn một nửa số lượng tổng các nút monitor để tránh các vấn đề split-brain mà các hệ thống khác đã gặp phải. Đây là lý do tại sao số lẻ các monitor được khuyến nghị. Trong số tất cả các monitor trong cụm, một trong số chúng hoạt động như là người lãnh đạo (leader). Các nút monitor khác có quyền trở thành người lãnh đạo nếu nút monitor đang là người lãnh đạo không khả dụng. Một cụm sản xuất phải có ít nhất ba nút monitor để đảm bảo tính sẵn có cao.

1. MDS

Ceph MDS (Metadata Server) là một thành phần của Ceph được sử dụng chỉ cho Ceph filesystem (CephFS) và các phương pháp lưu trữ khối khác; lưu trữ dựa trên object không đòi hỏi dịch vụ MDS. Ceph MDS hoạt động như một tiến trình nền, cho phép khách hàng mount một hệ thống tệp POSIX có kích thước bất kỳ. MDS không cung cấp dữ liệu trực tiếp cho khách hàng; việc cung cấp dữ liệu được thực hiện bởi OSD. MDS cung cấp một hệ thống tệp chia sẻ nhất quán với một lớp caching thông minh, giảm đáng kể việc đọc và ghi. MDS cũng hỗ trợ phân vùng cây con động và sử dụng một MDS duy nhất cho một phần metadata. Nó có tính động; các daemon có thể tham gia và rời đi, và quá trình tiếp quản các nút bị lỗi diễn ra nhanh chóng.

Hiện tại, MDS là thành phần duy nhất của Ceph chưa hoàn thiện cho môi trường sản xuất; hiện tại chỉ hỗ trợ một MDS và không thể mở rộng. Hiện đang có nhiều công việc Q&A để làm cho nó hoàn thiện cho môi trường sản xuất; chúng ta có thể mong đợi một số tin tức sắp tới.

MDS không lưu trữ dữ liệu cục bộ, điều này rất hữu ích trong một số tình huống. Nếu một tiến trình MDS bị lỗi, chúng ta có thể khởi động lại trên bất kỳ hệ thống nào có quyền truy cập vào cụm. Các tiến trình MDS được cấu hình là active và passive. MDS chính trở thành active, và các tiến trình khác sẽ chuyển sang chế độ standby. Trong trường hợp MDS chính gặp sự cố, nút thứ hai sẽ tiếp quản và được thăng chức thành active. Để khôi phục nhanh hơn, có thể chỉ định cho một nút standby theo dõi một trong các nút active, điều này sẽ giữ cùng dữ liệu trong bộ nhớ để tiền lưu trữ cache.

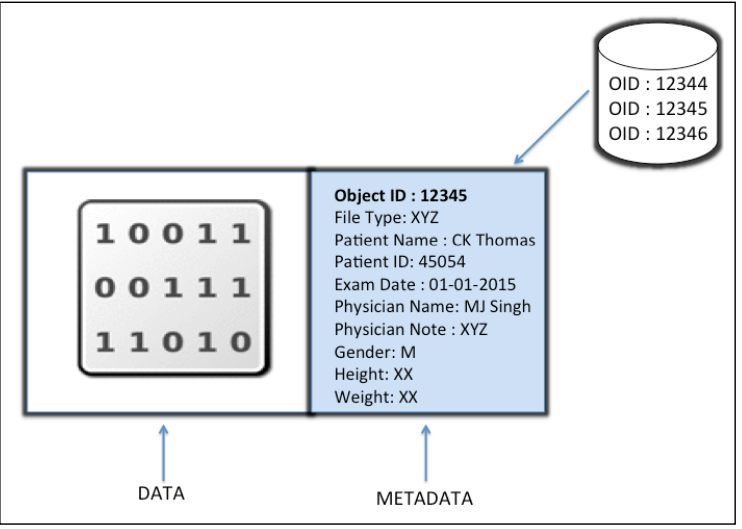
CHAPTER 4

Object

Một object thường bao gồm các thành phần dữ liệu và metadata được đóng gói cùng nhau và được cung cấp với một ID duy nhất. ID duy nhất đảm bảo rằng không có object nào khác có cùng ID object trong toàn bộ cụm lưu trữ và do đó đảm bảo tính duy nhất của object.

Khác với lưu trữ dựa trên tệp, trong đó các tệp bị giới hạn bởi kích thước, các object có thể có kích thước rất lớn cùng với metadata có kích thước thay đổi. Trong một object, dữ liệu được lưu trữ kèm theo metadata phong phú, chứa thông tin về ngữ cảnh và nội dung của dữ liệu. Metadata của lưu trữ object cho phép người dùng quản lý và truy cập vào dữ liệu không cấu trúc một cách chính xác.

Hãy xem xét ví dụ sau về việc lưu trữ hồ sơ bệnh nhân dưới dạng một object.



Một đối tượng không bị giới hạn bởi bất kỳ loại hoặc số lượng siêu dữ liệu nào; nó cho phép linh hoạt thêm một loại tùy chỉnh vào siêu dữ liệu và do đó hoàn toàn sở hữu dữ liệu của mình. Nó không sử dụng cấu trúc cây hoặc cấu trúc thư mục để lưu trữ; thay vào đó, nó được lưu trữ trong một không gian địa chỉ phẳng chứa hàng tỷ đối tượng mà không có bất kỳ sự phức tạp nào. Các đối tượng có thể được lưu trữ cục bộ hoặc chúng có thể được phân tách địa lý trong một không gian địa chỉ phẳng, tức là trong một không gian lưu trữ liên tục. Cơ chế lưu trữ này giúp các đối tượng đại diện duy nhất cho chính chúng trong toàn bộ cụm. Bất kỳ ứng dụng nào cũng có thể truy xuất dữ liệu từ một đối tượng dựa trên ID đối tượng của nó thông qua việc sử dụng các cuộc gọi API RESTful. Tương tự như cách URL hoạt động trên Internet, một ID đối tượng là một con trỏ duy nhất đến đối tượng của nó. Những đối tượng này được lưu trữ trong Các Thiết Bị Lưu Trữ Dựa Trên Đối Tượng (OSDs) theo cách được nhân bản, cung cấp tính sẵn có cao. Khi cụm lưu trữ Ceph nhận yêu cầu ghi dữ liệu từ các máy khách, nó lưu trữ dữ liệu dưới dạng các đối tượng. Tiến trình OSD sau đó viết dữ liệu vào một tệp trong hệ thống tệp OSD.

**CRUSH**

Trong ba thập kỷ qua, các cơ chế lưu trữ đã liên quan đến việc lưu trữ dữ liệu và siêu dữ liệu của nó. Siêu dữ liệu, tức là dữ liệu về dữ liệu, lưu trữ thông tin như nơi dữ liệu thực sự được lưu trữ trong một loạt các nút lưu trữ và mảng đĩa. Mỗi khi dữ liệu mới được thêm vào hệ thống lưu trữ, siêu dữ liệu của nó sẽ được cập nhật trước với vị trí vật lý nơi dữ liệu sẽ được lưu trữ, sau đó dữ liệu thực tế sẽ được lưu trữ. Quy trình này đã được chứng minh là hoạt động tốt khi chúng ta có một kích thước lưu trữ thấp trong khoảng từ gigabyte đến một vài terabyte dữ liệu, nhưng làm thế nào về việc lưu trữ dữ liệu cấp petabyte hoặc exabyte? Cơ chế này chắc chắn sẽ không phù hợp cho việc lưu trữ trong tương lai. Hơn nữa, nó tạo ra một điểm thất bại duy nhất cho hệ thống lưu trữ của bạn. Thật không may, nếu bạn mất siêu dữ liệu lưu trữ, bạn sẽ mất toàn bộ dữ liệu của bạn. Vì vậy, việc giữ siêu dữ liệu trung tâm an toàn khỏi thảm họa là rất quan trọng, bằng bất kỳ cách nào, có thể là việc giữ nhiều bản sao trên một nút duy nhất hoặc nhân bản toàn bộ dữ liệu và siêu dữ liệu để đạt được mức độ chịu lỗi cao hơn. Quản lý phức tạp như vậy của siêu dữ liệu là một rào cản đối với khả năng mở rộng, sẵn có cao và hiệu suất của hệ thống lưu trữ. Ceph là một cuộc cách mạng khi nói đến lưu trữ và quản lý dữ liệu. Nó sử dụng thuật toán Controlled Replication Under Scalable Hashing (CRUSH), cơ chế phân phối thông minh dữ liệu của Ceph. Thuật toán CRUSH là một trong những thành tựu quan trọng nhất của Ceph; nó là cốt lõi của toàn bộ cơ chế lưu trữ dữ liệu của Ceph. Khác với các hệ thống truyền thống dựa trên việc lưu trữ và quản lý một bảng siêu dữ liệu / chỉ mục trung tâm, Ceph sử dụng thuật toán CRUSH để tính toán một cách xác định nơi dữ liệu nên được ghi hoặc đọc từ. Thay vì lưu trữ siêu dữ liệu, CRUSH tính toán siêu dữ liệu khi có yêu cầu, do đó loại bỏ tất cả các hạn chế gặp phải khi lưu trữ siêu dữ liệu theo cách truyền thống.

**The CRUSH lookup**

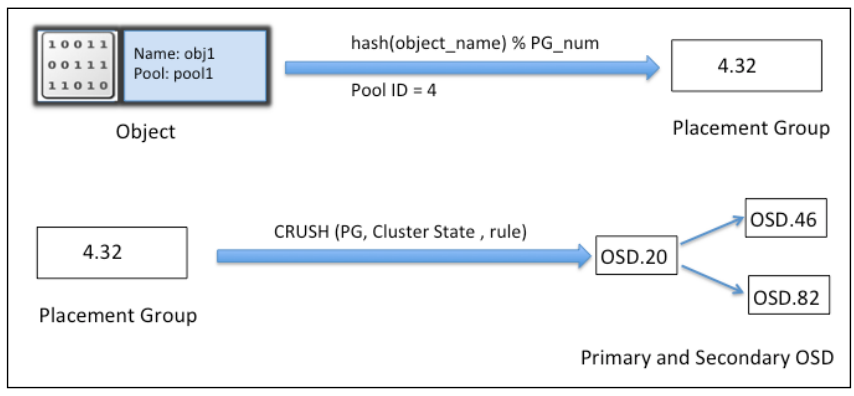
CRUSH (Controlled Replication Under Scalable Hashing) là một cơ chế trong Ceph cho phép tính toán metadata (siêu dữ liệu) chỉ khi cần thiết. Quá trình tính toán metadata cũng được gọi là CRUSH lookup, và các thiết bị máy tính hiện đại đã đủ mạnh mẽ để thực hiện các hoạt động CRUSH lookup một cách nhanh chóng và hiệu quả. Điều đặc biệt về CRUSH lookup là nó không phụ thuộc vào hệ thống cụ thể. Ceph cung cấp đủ linh hoạt cho các client để thực hiện tính toán metadata theo yêu cầu, tức là thực hiện CRUSH lookup bằng tài nguyên hệ thống riêng của mình, từ đó loại bỏ việc tìm kiếm trung tâm.

Trong quá trình đọc và ghi dữ liệu vào các cụm Ceph, các client trước tiên liên hệ với một Ceph monitor và nhận một bản sao của bản đồ cụm. Bản đồ cụm giúp client biết trạng thái và cấu hình của cụm Ceph. Dữ liệu được chuyển đổi thành các đối tượng với tên/ID đối tượng và pool. Đối tượng sau đó được băm với số nhóm đặt chỗ để tạo ra một nhóm đặt chỗ cuối cùng trong pool Ceph yêu cầu. Nhóm đặt chỗ được tính toán sau đó đi qua một CRUSH lookup để xác định vị trí OSD chính để lưu trữ hoặc truy xuất dữ liệu. Sau khi tính toán được ID OSD chính chính xác, client liên hệ trực tiếp với OSD này và lưu trữ dữ liệu. Tất cả các hoạt động tính toán này được thực hiện bởi các client, do đó không ảnh hưởng đến hiệu suất của cụm. Sau khi dữ liệu được ghi vào OSD chính, cùng một node thực hiện một hoạt động CRUSH lookup và tính toán vị trí cho các nhóm đặt chỗ và OSD phụ để sao lưu dữ liệu trên các cụm để đảm bảo khả năng sẵn có cao.

Xem xét ví dụ sau về CRUSH lookup và việc đặt đối tượng vào OSD:

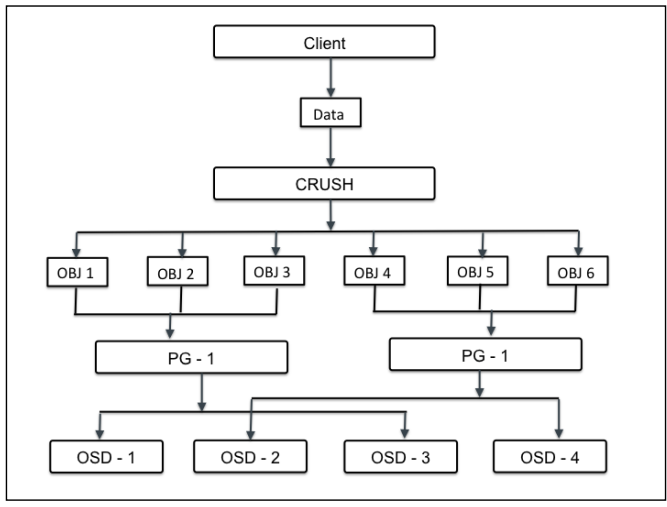
Trước hết, tên đối tượng và số nhóm đặt chỗ trong cụm được áp dụng với hàm băm và dựa trên các ID pool, một ID nhóm đặt chỗ (PGID) được tạo ra.

Tiếp theo, một CRUSH lookup được thực hiện trên PGID này để tìm ra OSD chính và OSD phụ để ghi dữ liệu.



**Placement groups**

Khi một cụm Ceph nhận yêu cầu lưu trữ dữ liệu, nó được chia thành các phần gọi là nhóm đặt chỗ (placement group - PG). Tuy nhiên, dữ liệu CRUSH được chia thành một tập hợp các đối tượng, và dựa trên hoạt động băm trên tên đối tượng, mức độ sao chép và tổng số nhóm đặt chỗ trong hệ thống, các ID nhóm đặt chỗ được tạo ra. Một nhóm đặt chỗ là một tập hợp logic của các đối tượng được sao chép trên các OSD để cung cấp tính tin cậy trong hệ thống lưu trữ. Tùy thuộc vào mức độ sao chép của pool Ceph của bạn, mỗi nhóm đặt chỗ được sao chép và phân phối trên nhiều OSD của một cụm Ceph. Bạn có thể coi một nhóm đặt chỗ như một khối logic chứa nhiều đối tượng sao cho khối logic này được ánh xạ vào nhiều OSD. Các nhóm đặt chỗ là yếu tố quan trọng cho tính mở rộng và hiệu suất của một hệ thống lưu trữ Ceph. Nếu không có nhóm đặt chỗ, việc quản lý và theo dõi hàng triệu đối tượng được sao chép và phân tán trên hàng trăm OSD sẽ trở nên khó khăn. Quản lý các đối tượng này mà không có nhóm đặt chỗ cũng sẽ dẫn đến một khoản phạt tính toán. Thay vì quản lý từng đối tượng một cách riêng lẻ, hệ thống phải quản lý các nhóm đặt chỗ với hàng loạt đối tượng. Điều này khiến cho Ceph trở thành một chức năng dễ quản lý hơn và ít phức tạp hơn. Mỗi nhóm đặt chỗ yêu cầu một lượng tài nguyên hệ thống, CPU và bộ nhớ nhất định vì mỗi nhóm đặt chỗ phải quản lý nhiều đối tượng. Số lượng nhóm đặt chỗ trong một cụm cần được tính toán một cách cẩn thận. Thông thường, việc tăng số lượng nhóm đặt chỗ trong cụm giảm tải trên mỗi OSD, nhưng việc tăng lên luôn phải được thực hiện một cách điều chỉnh. Đề nghị từ 50 đến 100 nhóm đặt chỗ mỗi OSD. Điều này để tránh sử dụng tài nguyên cao từ một nút OSD. Khi nhu cầu dữ liệu của bạn tăng lên, bạn sẽ cần mở rộng cụm của mình bằng cách điều chỉnh số lượng nhóm đặt chỗ. Khi các thiết bị được thêm hoặc loại bỏ khỏi một cụm, hầu hết các nhóm đặt chỗ vẫn ở vị trí của chúng; CRUSH quản lý việc di chuyển các nhóm đặt chỗ trên các cụm khác nhau.



**CEPH POOL**

Khái niệm về pool không mới trong hệ thống lưu trữ. Các hệ thống lưu trữ doanh nghiệp được quản lý bằng cách tạo ra nhiều pool; Ceph cũng cung cấp việc quản lý lưu trữ dễ dàng thông qua các pool lưu trữ. Một pool Ceph là một phân vùng logic để lưu trữ các đối tượng. Mỗi pool trong Ceph chứa một số nhóm đặt chỗ, mà trong đó chứa một số đối tượng được ánh xạ vào các OSD trên các cụm. Do đó, mỗi pool đơn lẻ được phân phối trên các nút cụm, và điều này cung cấp tính chống chịu. Việc triển khai Ceph ban đầu tạo ra một pool mặc định dựa trên yêu cầu của bạn; được khuyến nghị rằng bạn nên tạo ra các pool khác ngoài pool mặc định.

Một pool đảm bảo tính sẵn có dữ liệu bằng cách tạo ra số lượng bản sao đối tượng mong muốn, tức là bản sao hoặc mã hóa mất mát. Tính năng mã hóa mất mát (erasure coding - EC) đã được thêm vào Ceph, bắt đầu từ phiên bản Ceph Firefly. Mã hóa mất mát là một phương pháp bảo vệ dữ liệu trong đó dữ liệu được chia thành các mảnh, được mã hóa và sau đó được lưu trữ theo cách phân tán. Với tính phân tán của mình, Ceph sử dụng EC một cách xuất sắc.