Hierarchical Topics

DRAFT PROPOSAL

The following is a proposal for making kafka topics hierarchical. No one is currently working on this (that I know of) but it seemed like a good idea to get the idea down on paper and throw it out there.

Motivation

LinkedIn uses Kafka very heavily and we have noticed two problems that arise due to the use of a flat, per-cluster namespace for topics:

1. There is no logical grouping of topics. Topics often group together naturally by the application area (ads, search, etc). With a flat namespace this grouping is all implicit based on the naming conventions. As we have grown to hundreds of topics this has become a challenge.
2. There are a large number of clusters at LinkedIn. Finding the right cluster to connect to for the data stream you need can be a bit confusing. Our clusters are organized in the following way:
    a. Currently we keep a cluster for each major usage paradigm:
        i. tracking data
        ii. metrics/operational data
        iii. queuing
        iv. outbound data deployments from hadoop
    b. But the above pattern is replicated for each live serving data center, plus aggregate clusters that unify the full data feed and replicas in the offline processing data centers.

The result of the above is that finding the right topic and cluster to subscribe to is a bit of a challenge even now. As we use kafka topics for more ad hoc processing use cases the expectation is that the number of topics will increase further.

This proposal attempts to create a more logical overview of all this data decoupled from the physical clusters.

Proposal

At its heart the idea is to allow two things:

1. Add directories. So a consumer might subscribe to /tracking/search/search_click_events instead of just search_click_event.
2. Add mount points that allow unifying the namespace across clusters. Each cluster would mount into a location in the global namespace, and provide its topics there. For example for geographical replication we could have a directory hierarchy with a top-level directory for each cluster /la/tracking might be the mount point for the tracking cluster in the la data center. This allows us to abstract over the physical mapping of topics to clusters (today we might be running /tracking as a separate cluster from /metrics, but tomorrow if we had better multi-tenancy features we might decide to combine them into the same cluster, regardless the "URL" is the same.

Notes

I have not thought through the details of an implementation. Here are a few basic thoughts, though:

- Arguably we might want to have a single namespace per data center rather than a single global namespace, since in our architecture we want people to always connect to the local cluster.
- The zookeeper data structures and on-disk structures need not change. That is the directory structure in zk or on disk could remain flat. We would need to replace '/' with some out of band character to prevent name collisions. Since we are planning to remove direct zk access from the client zk structures are an implementation detail so any convention should be fine.
- There would still be one zk per colo. This does not require a single zk for everything.
- Initially we would not support wild cards that cross volumes. E.g. subscribing to '/*' wouldn't subscribe to all topics on all clusters. This would require explict consumer support.
- Each cluster would be configured to know about it's direct parents and children as well as its mount point url. Clients can initially connect to any cluster to derive the location of a topic by traversing up or down the directory tree. This is better than having the client always begin its search at '/' because it would mean that the local cluster's paths are always available even if the '/' cluster is unavailable.