KIP-64 -Allow underlying distributed filesystem to take over replication depending on configuration

Status

Current state: [Under discussion]

Discussion thread: here

JIRA: KIP-64 -Allow underlying distributed filesystem to take over replication depending on configuration

Please keep the discussion on the mailing list rather than commenting on the wiki (wiki discussions get unwieldy fast).

Motivation

Distributed data stores can be vastly improved by integrating with Kafka. Some of these improvements are:

1. They can participate easily in the whole Kafka ecosystem
2. Data ingesting speeds can be improved

Distributed data stores come with their own replication. Kafka replication is a duplication of functionality for them.Kafka should defer replication to underlying file system if the configuration mandates it.

In the newly added configuration a flush to the filesystem should consider a signal that the message is replicated.

Public Interfaces

A new configuration entry which tells the broker that it is running on a distributed storage.

Proposed Changes

Code changes:

1. A new configuration entry which tells the broker that it is running on a distributed storage.
2. When this configuration is set:
    a. Replicas do not run the replication code
    b. Whenever flush returns on a log, all the other brokers in the system are marked as part of ISR

Changes in behavior:

1. No replication traffic between the brokers at Kafka level
2. Async replication behavior(request.required.acks=0 or 1) does not change
3. Replication factor is ignored / taken care by the underlying storage
4. After flush all the brokers in the system are marked as in-sync replicas
5. request.timeout.ms needs to be greater than log.flush.interval.ms to ensure that unncessary retires do not happen.

Deployment changes:

1. Distributed storage is mounted (via NFS/SMB) on log dirs

Compatibility, Deprecation, and Migration Plan

The behavior kicks in only after this particular flag is set. Otherwise the existing code works as is.

Rejected Alternatives

1. Keep the existing replication functionality. This is rejected because:
    a. Duplication of functionality
    b. Perf deterioration
    c. Unnecessary copies of data
2. Have this as a per topic configuration. This is rejected because:
    a. Hybrid deployment is not anticipated. Folks will use deployments with either replicated storage OR without it.