

KIP-699: Update FindCoordinator to resolve multiple Coordinators at a time

- [Status](#)
- [Motivation](#)
- [Public Interfaces](#)
 - [FindCoordinators API](#)
- [Proposed Changes](#)
- [Compatibility, Deprecation, and Migration Plan](#)
- [Rejected Alternatives](#)

Status

Current state: Accepted

Discussion thread: [here](#) [Change the link from the KIP proposal email archive to your own email thread]

JIRA: [here](#) [Change the link from KAFKA-1 to your own ticket]

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

Motivation

FindCoordinator only allows to look up the coordinator for a single resource. For use cases working with several consumer groups, this forces to send one FindCoordinator request for each group. This can have a negative impact on performance.

This is the case with the following Admin APIs:

- `deleteConsumerGroups()`
- `describeConsumerGroups()`

This also affects the new API proposed in [KIP-709: Extend OffsetFetch requests to accept multiple group ids](#).

These APIs take a collection of groups as input. Being able to find coordinators in batches would allow to use a single find coordinator request to resolve them all instead of 1 request per group as of today.

Public Interfaces

We'll rename the FindCoordinator message to FindCoordinators

FindCoordinators API

Bump FindCoordinatorsRequest version to 4 and include a new field "Keys". The existing "Key" field will be limited to versions less than 4. As we don't have use cases for finding coordinators for both groups and transactions at the same time, the "KeyType" field is kept and specifies the request type for all keys in "Keys".

```

{
  "apiKey": 10,
  "type": "request",
  "name": "FindCoordinatorsRequest",
  // Version 1 adds KeyType.
  //
  // Version 2 is the same as version 1.
  //
  // Version 3 is the first flexible version.
  //
  // Version 4 adds support for batching via Keys
  "validVersions": "0-4",
  "flexibleVersions": "3+",
  "fields": [
    { "name": "Key", "type": "string", "versions": "0-3",
      "about": "The coordinator key." },
    { "name": "KeyType", "type": "int8", "versions": "1+", "default": "0", "ignorable": false,
      "about": "The coordinator key type. (Group, transaction, etc.)" },
    { "name": "Keys", "type": "[]string", "versions": "4+",
      "about": "The coordinator keys." }
  ]
}

```

Bump FindCoordinatorsResponse version to 4 and include a new field "Coordinators".

```

{
  "apiKey": 10,
  "type": "response",
  "name": "FindCoordinatorsResponse",
  // Version 1 adds throttle time and error messages.
  //
  // Starting in version 2, on quota violation, brokers send out responses before throttling.
  //
  // Version 3 is the first flexible version.
  //
  // Version 4 adds support for batching via Coordinators
  "validVersions": "0-4",
  "flexibleVersions": "3+",
  "fields": [
    { "name": "ThrottleTimeMs", "type": "int32", "versions": "1+", "ignorable": true,
      "about": "The duration in milliseconds for which the request was throttled due to a quota violation, or
zero if the request did not violate any quota." },
    { "name": "ErrorCode", "type": "int16", "versions": "0-3",
      "about": "The error code, or 0 if there was no error." },
    { "name": "ErrorMessage", "type": "string", "versions": "1-3", "nullableVersions": "1-3", "ignorable": true,
      "about": "The error message, or null if there was no error." },
    { "name": "NodeId", "type": "int32", "versions": "0-3", "entityType": "brokerId",
      "about": "The node id." },
    { "name": "Host", "type": "string", "versions": "0-3",
      "about": "The host name." },
    { "name": "Port", "type": "int32", "versions": "0-3",
      "about": "The port." },
    { "name": "Coordinators", "type": "[]Coordinator", "versions": "4+", "about": "Each coordinator result in
the response", "fields": [
      { "name": "Key", "type": "string", "versions": "4+", "about": "The coordinator key." },
      { "name": "NodeId", "type": "int32", "versions": "4+", "entityType": "brokerId",
        "about": "The node id." },
      { "name": "Host", "type": "string", "versions": "4+", "about": "The host name." },
      { "name": "Port", "type": "int32", "versions": "4+", "about": "The port." },
      { "name": "ErrorCode", "type": "int16", "versions": "4+",
        "about": "The error code, or 0 if there was no error." },
      { "name": "ErrorMessage", "type": "string", "versions": "4+", "nullableVersions": "4+", "ignorable": true,
        "about": "The error message, or null if there was no error." }
    ]
  ]
}

```

Proposed Changes

KafkaConsumer, KafkaProducer and KafkaAdminClient will be updated to use the new version of FindCoordinators. If brokers don't support v4, clients will automatically revert to the old behaviour.

KafkaAdminClient will be updated to find all group coordinators in a single call if possible for both deleteConsumerGroups() and describeConsumerGroups(). If that's not possible, it will revert to send a FindCoordinators request for each group.

A new exception "NoBatchedFindCoordinatorsException" will be introduced:

```
package org.apache.kafka.common.errors;

/**
 * Indicates that it is not possible to lookup coordinators in batches with FindCoordinator. Instead
 * coordinators must be looked up one by one.
 */
public class NoBatchedFindCoordinatorsException extends UnsupportedOperationException {
    private static final long serialVersionUID = 1L;

    public NoBatchedFindCoordinatorsException(String message, Throwable cause) {
        super(message, cause);
    }

    public NoBatchedFindCoordinatorsException(String message) {
        super(message);
    }
}
```

This will be thrown by FindCoordinatorRequest when attempting to use v4 or above on brokers that don't support it.

Compatibility, Deprecation, and Migration Plan

If this feature is not supported by brokers, clients will automatically revert to the old behaviour. There are no public API changes.

Rejected Alternatives

None