

KIP-888: Batch describe ACLs and describe client quotas

- [Status](#)
- [Motivation](#)
- [Public Interfaces](#)
 - [Update DescribeACLs Request/Response](#)
 - [Update DescribeClientQuotas Request/Response](#)
 - [New Admin APIs](#)
 - [Deprecate the existing describeACLs and describeClientQuotas methods](#)
 - [Update DescribeClientQuotasResult and DescribeACLsResult](#)
- [Proposed Changes](#)
- [Compatibility, Deprecation, and Migration Plan](#)
- [Test Plan](#)
- [Rejected Alternatives](#)

Status

Current state: Under Discussion

Discussion thread: [Thread](#)

JIRA:

 Unable to render Jira issues macro, execution error.

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

Motivation

Currently the DescribeACLs and DescribeClientQuotas APIs only allow to retrieve ACLs/quotas for a single user at a time. Tooling managing users often have to update several users at a time, this requires making multiple calls to the Admin API which result in multiple requests to the cluster. Being able to retrieve data from multiple users at a time would simplify client code and reduce the amount of requests necessary to make ACL and quota changes.

Public Interfaces

Update DescribeACLs Request/Response

With version 4, we are adding the Filters field to DescribeAclsRequest.

```

{
  "apiKey": 29,
  "type": "request",
  "listeners": ["zkBroker", "broker", "controller"],
  "name": "DescribeAclsRequest",
  // Version 1 adds resource pattern type.
  // Version 2 enables flexible versions.
  // Version 3 adds user resource type.
  // Version 4 adds support for batching
  "validVersions": "0-4",
  "flexibleVersions": "2+",
  "fields": [
    ...
    <existing fields>
    ...
    { "name": "Filters",
      "type": "[]DescribeAclsFilter",
      "versions": "4+",
      "about": "The filters to use when describing ACLs.",
      "fields": [
        { "name": "ResourceTypeFilter", "type": "int8", "versions": "4+",
          "about": "The resource type." },
        { "name": "ResourceNameFilter", "type": "string", "versions": "4+", "nullableVersions": "4+",
          "about": "The resource name." },
        { "name": "PatternTypeFilter", "type": "int8", "versions": "4+", "default": "3", "ignorable": false,
          "about": "The pattern type." },
        { "name": "PrincipalFilter", "type": "string", "versions": "4+", "nullableVersions": "4+",
          "about": "The principal filter, or null to accept all principals." },
        { "name": "HostFilter", "type": "string", "versions": "4+", "nullableVersions": "4+",
          "about": "The host filter, or null to accept all hosts." },
        { "name": "Operation", "type": "int8", "versions": "4+",
          "about": "The ACL operation." },
        { "name": "PermissionType", "type": "int8", "versions": "4+",
          "about": "The permission type." }
      ]
    }
  ]
}

```

With version 4, we are adding the FilterResults field to DescribeAclsResponse. We also have to rename the type of the existing Resources and Acls fields to avoid them conflicting with the new fields.

```

{
  "apiKey": 29,
  "type": "response",
  "name": "DescribeAclsResponse",
  // Version 1 adds PatternType.
  // Starting in version 1, on quota violation, brokers send out responses before throttling.
  // Version 2 enables flexible versions.
  // Version 3 adds user resource type.
  // Version 4 adds support for batching.
  "validVersions": "0-4",
  "flexibleVersions": "2+",
  "fields": [
    { "name": "ThrottleTimeMs", "type": "int32", "versions": "0+",
      "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": "0-3", "nullableVersions": "0-3",
      "about": "The error message, or null if there was no error." },
    { "name": "Resources", "type": "[]OldDescribeAclsResource", "versions": "0-3",
      "about": "Each Resource that is referenced in an ACL.", "fields": [
        { "name": "ResourceType", "type": "int8", "versions": "0-3",
          "about": "The resource type." },
        { "name": "ResourceName", "type": "string", "versions": "0-3",
          "about": "The resource name." },
        { "name": "PatternType", "type": "int8", "versions": "1-3", "default": "3", "ignorable": false,

```

```

        "about": "The resource pattern type." },
    { "name": "Acls", "type": "[]OldAclDescription", "versions": "0-3",
      "about": "The ACLs.", "fields": [
        { "name": "Principal", "type": "string", "versions": "0-3",
          "about": "The ACL principal." },
        { "name": "Host", "type": "string", "versions": "0-3",
          "about": "The ACL host." },
        { "name": "Operation", "type": "int8", "versions": "0-3",
          "about": "The ACL operation." },
        { "name": "PermissionType", "type": "int8", "versions": "0-3",
          "about": "The ACL permission type." }
      ]
    }
  ],
  {
    "name": "FilterResults",
    "type": "[]DescribeAclsFilterResult",
    "versions": "4+",
    "about": "The results for each filter.",
    "fields": [
      { "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+",
        "about": "The error message, or null if there was no error." },
      { "name": "Resources", "type": "[]DescribeAclsResource", "versions": "4+",
        "about": "Each Resource that is referenced in an ACL.", "fields": [
          { "name": "ResourceType", "type": "int8", "versions": "4+",
            "about": "The resource type." },
          { "name": "ResourceName", "type": "string", "versions": "4+",
            "about": "The resource name." },
          { "name": "PatternType", "type": "int8", "versions": "4+", "default": "3", "ignorable": false,
            "about": "The resource pattern type." },
          { "name": "Acls", "type": "[]AclDescription", "versions": "4+",
            "about": "The ACLs.", "fields": [
              { "name": "Principal", "type": "string", "versions": "4+",
                "about": "The ACL principal." },
              { "name": "Host", "type": "string", "versions": "4+",
                "about": "The ACL host." },
              { "name": "Operation", "type": "int8", "versions": "4+",
                "about": "The ACL operation." },
              { "name": "PermissionType", "type": "int8", "versions": "4+",
                "about": "The ACL permission type." }
            ]
          }
        ]
      ]
    }
  ]
}

```

Update DescribeClientQuotas Request/Response

With version 2 we are adding the Filters field to DescribeClientQuotasRequest. We also have to rename the type of the existing Components fields to avoid them conflicting with the new fields.

```

{
  "apiKey": 48,
  "type": "request",
  "listeners": ["zkBroker", "broker"],
  "name": "DescribeClientQuotasRequest",
  // Version 1 enables flexible versions.
  // Version 2 adds support for batching.
  "validVersions": "0-2",
  "flexibleVersions": "1+",
  "fields": [
    { "name": "Components", "type": "[]OldComponentData", "versions": "0-1",
      "about": "Filter components to apply to quota entities.", "fields": [
        { "name": "EntityType", "type": "string", "versions": "0-1",
          "about": "The entity type that the filter component applies to." },
        { "name": "MatchType", "type": "int8", "versions": "0-1",
          "about": "How to match the entity {0 = exact name, 1 = default name, 2 = any specified name}." },
        { "name": "Match", "type": "string", "versions": "0-1", "nullableVersions": "0-1",
          "about": "The string to match against, or null if unused for the match type." }
      ]
    },
    { "name": "Strict", "type": "bool", "versions": "0+",
      "about": "Whether the match is strict, i.e. should exclude entities with unspecified entity types." },
    {
      "name": "Filters",
      "type": "[]DescribeClientQuotasFilter",
      "versions": "2+",
      "about": "The filters to use when describing client quotas.",
      "fields": [
        { "name": "Components", "type": "[]ComponentData", "versions": "2+",
          "about": "Filter components to apply to quota entities.", "fields": [
            { "name": "EntityType", "type": "string", "versions": "2+",
              "about": "The entity type that the filter component applies to." },
            { "name": "MatchType", "type": "int8", "versions": "2+",
              "about": "How to match the entity {0 = exact name, 1 = default name, 2 = any specified name}." },
            { "name": "Match", "type": "string", "versions": "2+", "nullableVersions": "2+",
              "about": "The string to match against, or null if unused for the match type." }
          ]
        }
      ]
    }
  ]
}

```

With version 2 we are adding the Filters field to DescribeClientQuotasResponse. We also have to rename the type of the existing Entries, Entities and Values fields to avoid them conflicting with the new fields.

```

{
  "apiKey": 48,
  "type": "response",
  "name": "DescribeClientQuotasResponse",
  // Version 1 enables flexible versions.
  // Version 2 adds support for batching.
  "validVersions": "0-2",
  "flexibleVersions": "1+",
  "fields": [
    { "name": "ThrottleTimeMs", "type": "int32", "versions": "0+",
      "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": "0-1",
      "about": "The error code, or `0` if the quota description succeeded." },
    { "name": "ErrorMessage", "type": "string", "versions": "0-1", "nullableVersions": "0-1",
      "about": "The error message, or `null` if the quota description succeeded." },
    { "name": "Entries", "type": "[]OldEntryData", "versions": "0-1", "nullableVersions": "0-1",
      "about": "A result entry.", "fields": [
        { "name": "Entity", "type": "[]OldEntityData", "versions": "0-1",
          "about": "The quota entity description.", "fields": [
            { "name": "EntityType", "type": "string", "versions": "0-1",
              "about": "The entity type." },
            { "name": "EntityName", "type": "string", "versions": "0-1", "nullableVersions": "0-1",
              "about": "The entity name, or null if the default." }
          ]},
        { "name": "Values", "type": "[]OldValueData", "versions": "0-1",
          "about": "The quota values for the entity.", "fields": [
            { "name": "Key", "type": "string", "versions": "0-1",
              "about": "The quota configuration key." },
            { "name": "Value", "type": "float64", "versions": "0-1",
              "about": "The quota configuration value." }
          ]}
      ]}
  ]},
  {
    "name": "FilterResults",
    "type": "[]DescribeClientQuotasFilterResult",
    "versions": "2+",
    "about": "The results for each filter.",
    "fields": [
      { "name": "ErrorCode", "type": "int16", "versions": "2+",
        "about": "The error code, or `0` if the quota description succeeded." },
      { "name": "ErrorMessage", "type": "string", "versions": "2+", "nullableVersions": "2+",
        "about": "The error message, or `null` if the quota description succeeded." },
      { "name": "Entries", "type": "[]EntryData", "versions": "2+", "nullableVersions": "2+",
        "about": "A result entry.", "fields": [
          { "name": "Entity", "type": "[]EntityData", "versions": "2+",
            "about": "The quota entity description.", "fields": [
              { "name": "EntityType", "type": "string", "versions": "2+",
                "about": "The entity type." },
              { "name": "EntityName", "type": "string", "versions": "2+", "nullableVersions": "2+",
                "about": "The entity name, or null if the default." }
            ]},
          { "name": "Values", "type": "[]ValueData", "versions": "2+",
            "about": "The quota values for the entity.", "fields": [
              { "name": "Key", "type": "string", "versions": "2+",
                "about": "The quota configuration key." },
              { "name": "Value", "type": "float64", "versions": "2+",
                "about": "The quota configuration value." }
            ]}
        ]}
      ]}
  ]}
}

```

New Admin APIs

Four new methods will be added to the Admin interface:

```

/**
 * This is a convenience method for {@link #describeAcls(Collection<AclBindingFilter>,
DescribeAclsOptions)} with
 * default options. See the overload for more details.
 * <p>
 *
 * This operation is supported by brokers with version 3.5 or higher.
 * @param filters The filters to use.
 * @return The DescribeAclsResult.
 */
default DescribeAclsResult describeAcls(Collection<AclBindingFilter> filters) {
    return describeAcls(filters, new DescribeAclsOptions());
}

/**
 * Lists access control lists (ACLs) according to the supplied filters.
 * <p>
 * Note: it may take some time for changes made by {@code createAcls} or {@code deleteAcls} to be reflected
 * in the output of {@code describeAcls}.
 * <p>
 *
 * This operation is supported by brokers with version 3.5 or higher.
 * @param filters The filters to use.
 * @param options The options to use when listing the ACLs.
 * @return The DescribeAclsResult.
 */
DescribeAclsResult describeAcls(Collection<AclBindingFilter> filters, DescribeAclsOptions options);

/**
 * Describes all entities matching the provided filters that have at least one client quota configuration
 * value defined.
 * <p>
 * This is a convenience method for {@link #describeClientQuotas(Collection<ClientQuotaFilter>,
DescribeClientQuotasOptions)}
 * with default options. See the overload for more details.
 * <p>
 * This operation is supported by brokers with version 3.5 or higher.
 *
 * @param filters The filters to apply to match entities.
 * @return The DescribeClientQuotasResult.
 */
default DescribeClientQuotasResult describeClientQuotas(Collection<ClientQuotaFilter> filters) {
    return describeClientQuotas(filters, new DescribeClientQuotasOptions());
}

/**
 * Describes all entities matching the provided filters that have at least one client quota configuration
 * value defined.
 * <p>
 * The following exceptions can be anticipated when calling {@code get()} on the future from the
 * returned {@link DescribeClientQuotasResult}:
 * <ul>
 * <li>{@link org.apache.kafka.common.errors.ClusterAuthorizationException}
 * If the authenticated user didn't have describe access to the cluster.</li>
 * <li>{@link org.apache.kafka.common.errors.InvalidRequestException}
 * If the request details are invalid. e.g., an invalid entity type was specified.</li>
 * <li>{@link org.apache.kafka.common.errors.TimeoutException}
 * If the request timed out before the describe could finish.</li>
 * </ul>
 * <p>
 * This operation is supported by brokers with version 3.5 or higher.
 *
 * @param filters The filters to apply to match entities.
 * @param options The options to use.
 * @return The DescribeClientQuotasResult.
 */
DescribeClient
QuotasResult describeClientQuotas(Collection<ClientQuotaFilter> filters, DescribeClientQuotasOptions options);

```

Deprecate the existing describeACLs and describeClientQuotas methods

Users should now use the new methods.

```
@Deprecated
default DescribeAclsResult describeAcls(AclBindingFilter filter) { ... }

@Deprecated
DescribeAclsResult describeAcls(AclBindingFilter filter, DescribeAclsOptions options);

@Deprecated
default DescribeClientQuotasResult describeClientQuotas(ClientQuotaFilter filter) { ... }

@Deprecated
DescribeClientQuotasResult describeClientQuotas(ClientQuotaFilter filter, DescribeClientQuotasOptions options);
```

Update DescribeClientQuotasResult and DescribeACLsResult

For DescribeClientQuotasResult: deprecate the existing constructor and values() method. Unfortunately we can't have another values() method so the new method is called allValues(). Also add the all() method.

DescribeACLsResult

```
@Deprecated
DescribeAclsResult(KafkaFuture<Collection<AclBinding>> future) { .. }

DescribeAclsResult(Map<AclBindingFilter, KafkaFuture<Collection<AclBinding>>> futures) { ... }

@Deprecated
public KafkaFuture<Collection<AclBinding>> values() { ... }

/**
 * Return a future containing the ACLs requested.
 */
public Map<AclBindingFilter, KafkaFuture<Collection<AclBinding>>> allValues() { ... }

/**
 * Return a future which succeeds only if all the ACL descriptions succeed.
 */
public KafkaFuture<Map<AclBindingFilter, Collection<AclBinding>>> all() { ... }
```

For DescribeClientQuotasResult: Deprecate the existing constructor and add a new non public constructor. Deprecate the entities method and add 2 methods: values() and all().

DescribeClientQuotasResult

```
@Deprecated
public DescribeClientQuotasResult(KafkaFuture<Map<ClientQuotaEntity, Map<String, Double>>> entities) { ... }

DescribeClientQuotasResult(Map<ClientQuotaFilter, KafkaFuture<Map<ClientQuotaEntity, Map<String, Double>>>>
futures) { ... }

/**
 * Returns a map from quota entity to a future which can be used to check the status of the operation.
 */
@Deprecated
public KafkaFuture<Map<ClientQuotaEntity, Map<String, Double>>> entities() { ... }

public Map<ClientQuotaFilter, KafkaFuture<Map<ClientQuotaEntity, Map<String, Double>>>> values() { ... }

/**
 * Return a future which succeeds only if all the quota descriptions succeed.
 */
public KafkaFuture<Map<ClientQuotaFilter, Map<ClientQuotaEntity, Map<String, Double>>>> all() { ... }
```


Proposed Changes

This KIP consists in:

- 1) Extending the wire protocol: It introduces new version for the DescribeACLs and DescribeClientQuotas APIs. In both cases the change is to allow passing a collection of entities instead of a single one. The older entity fields will only be accepted in requests/responses for older versions.
- 2) Extending the Admin API: It introduces four new methods to the Admin interface. These methods accept a collection of filters to retrieve ACLs and quotas for multiple entities in a single call. They follow the same pattern as existing calls that already accepts multiple entities such as deleteACLs(). Existing methods for single entities will be deprecated. When using the new methods, if the brokers don't support the new protocol, the Admin client will use the older API and send multiple requests if necessary.
- 3) Updating the Result types for describeACLs and describeClientQuotas: For consistency, it makes sense to keep the existing types. Existing methods will be deprecated and new methods will be added to handle collections of entities.

Compatibility, Deprecation, and Migration Plan

This is adding new APIs. The older describeACLs and describeClientQuotas calls will be kept but deprecated. When using brokers that don't support the new APIs, the Admin client will be able to downconvert requests and send multiple older requests to satisfy calls with several entities.

Test Plan

We will add unit and integration tests to cover the internal changes and new APIs.

Rejected Alternatives

None