package com.twitter.search.earlybird\_root;

import javax.inject.Inject;

import javax.inject.Named;

import com.twitter.search.common.decider.SearchDecider;

import com.twitter.search.common.metrics.SearchCounter;

import com.twitter.search.common.root.SearchRootModule;

import com.twitter.search.earlybird\_root.common.EarlybirdRequestType;

/\*\*

\* Determines if a root should send requests to certain partitions based on if they have been turned

\* off by decider.

\*/

public class PartitionAccessController {

private final String clusterName;

private final SearchDecider decider;

@Inject

public PartitionAccessController(

@Named(SearchRootModule.NAMED\_SEARCH\_ROOT\_NAME) String clusterName,

@Named(SearchRootModule.NAMED\_PARTITION\_DECIDER) SearchDecider partitionDecider) {

this.clusterName = clusterName;

this.decider = partitionDecider;

}

/\*\*

\* Should root send requests to a given partition

\* Designed to be used to quickly stop hitting a partition of there are problems with it.

\*/

public boolean canAccessPartition(

String tierName, int partitionNum, String clientId, EarlybirdRequestType requestType) {

String partitionDeciderName =

String.format("cluster\_%s\_skip\_tier\_%s\_partition\_%s", clusterName, tierName, partitionNum);

if (decider.isAvailable(partitionDeciderName)) {

SearchCounter.export(partitionDeciderName).increment();

return false;

}

String clientDeciderName = String.format("cluster\_%s\_skip\_tier\_%s\_partition\_%s\_client\_id\_%s",

clusterName, tierName, partitionNum, clientId);

if (decider.isAvailable(clientDeciderName)) {

SearchCounter.export(clientDeciderName).increment();

return false;

}

String requestTypeDeciderName = String.format(

"cluster\_%s\_skip\_tier\_%s\_partition\_%s\_request\_type\_%s",

clusterName, tierName, partitionNum, requestType.getNormalizedName());

if (decider.isAvailable(requestTypeDeciderName)) {

SearchCounter.export(requestTypeDeciderName).increment();

return false;

}

String clientRequestTypeDeciderName = String.format(

"cluster\_%s\_skip\_tier\_%s\_partition\_%s\_client\_id\_%s\_request\_type\_%s",

clusterName, tierName, partitionNum, clientId, requestType.getNormalizedName());

if (decider.isAvailable(clientRequestTypeDeciderName)) {

SearchCounter.export(clientRequestTypeDeciderName).increment();

return false;

}

return true;

}

public String getClusterName() {

return clusterName;

}

}