package com.twitter.search.earlybird.factory;

import java.io.IOException;

import com.google.common.base.Preconditions;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import com.twitter.common.util.Clock;

import com.twitter.decider.Decider;

import com.twitter.search.common.aurora.AuroraInstanceKey;

import com.twitter.search.common.aurora.AuroraSchedulerClient;

import com.twitter.search.common.concurrent.ScheduledExecutorServiceFactory;

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

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

import com.twitter.search.common.util.ml.tensorflow\_engine.TensorflowModelsManager;

import com.twitter.search.common.util.zktrylock.ZooKeeperTryLockFactory;

import com.twitter.search.earlybird.EarlybirdDarkProxy;

import com.twitter.search.earlybird.EarlybirdFinagleServerManager;

import com.twitter.search.earlybird.EarlybirdFuturePoolManager;

import com.twitter.search.earlybird.EarlybirdIndexConfig;

import com.twitter.search.earlybird.EarlybirdServer;

import com.twitter.search.earlybird.EarlybirdServerSetManager;

import com.twitter.search.earlybird.EarlybirdWarmUpManager;

import com.twitter.search.earlybird.QualityFactor;

import com.twitter.search.earlybird.UpdateableEarlybirdStateManager;

import com.twitter.search.earlybird.archive.ArchiveEarlybirdIndexConfig;

import com.twitter.search.earlybird.common.config.EarlybirdConfig;

import com.twitter.search.earlybird.common.userupdates.UserScrubGeoMap;

import com.twitter.search.earlybird.common.userupdates.UserUpdatesChecker;

import com.twitter.search.earlybird.common.userupdates.UserTable;

import com.twitter.search.earlybird.exception.CriticalExceptionHandler;

import com.twitter.search.earlybird.index.EarlybirdSegmentFactory;

import com.twitter.search.earlybird.ml.ScoringModelsManager;

import com.twitter.search.earlybird.partition.AudioSpaceEventsStreamIndexer;

import com.twitter.search.earlybird.partition.AudioSpaceTable;

import com.twitter.search.earlybird.partition.DynamicPartitionConfig;

import com.twitter.search.earlybird.partition.EarlybirdStartup;

import com.twitter.search.earlybird.partition.MultiSegmentTermDictionaryManager;

import com.twitter.search.earlybird.partition.PartitionConfig;

import com.twitter.search.earlybird.partition.PartitionManager;

import com.twitter.search.earlybird.partition.SegmentManager;

import com.twitter.search.earlybird.partition.SegmentSyncConfig;

import com.twitter.search.earlybird.partition.UserScrubGeoEventStreamIndexer;

import com.twitter.search.earlybird.partition.UserUpdatesStreamIndexer;

import com.twitter.search.earlybird.querycache.QueryCacheConfig;

import com.twitter.search.earlybird.querycache.QueryCacheManager;

import com.twitter.search.earlybird.stats.EarlybirdSearcherStats;

import com.twitter.search.earlybird.util.TermCountMonitor;

import com.twitter.search.earlybird.util.TweetCountMonitor;

/\*\*

\* This is the wiring file that builds EarlybirdServers.

\* Production and test code share this same wiring file.

\* <p/>

\* To supply mocks for testing, one can do so by supplying a different

\* EarlybirdWiringModule to this wiring file.

\*/

public final class EarlybirdServerFactory {

private static final Logger LOG = LoggerFactory.getLogger(EarlybirdServerFactory.class);

/\*\*

\* Creates the EarlybirdServer based on the bindings in the given wire module.

\*

\* @param earlybirdWireModule The wire module that specifies all required bindings.

\*/

public EarlybirdServer makeEarlybirdServer(EarlybirdWireModule earlybirdWireModule)

throws IOException {

LOG.info("Started making an Earlybird server");

CriticalExceptionHandler criticalExceptionHandler = new CriticalExceptionHandler();

Decider decider = earlybirdWireModule.provideDecider();

SearchDecider searchDecider = new SearchDecider(decider);

EarlybirdWireModule.ZooKeeperClients zkClients = earlybirdWireModule.provideZooKeeperClients();

ZooKeeperTryLockFactory zkTryLockFactory =

zkClients.stateClient.createZooKeeperTryLockFactory();

EarlybirdIndexConfig earlybirdIndexConfig =

earlybirdWireModule.provideEarlybirdIndexConfig(

decider, earlybirdWireModule.provideSearchIndexingMetricSet(),

criticalExceptionHandler);

SearchStatsReceiver earlybirdServerStats =

earlybirdWireModule.provideEarlybirdServerStatsReceiver();

EarlybirdSearcherStats tweetsSearcherStats =

earlybirdWireModule.provideTweetsSearcherStats();

DynamicPartitionConfig dynamicPartitionConfig =

earlybirdWireModule.provideDynamicPartitionConfig();

PartitionConfig partitionConfig = dynamicPartitionConfig.getCurrentPartitionConfig();

LOG.info("Partition config info [Cluster: {}, Tier: {}, Partition: {}, Replica: {}]",

partitionConfig.getClusterName(),

partitionConfig.getTierName(),

partitionConfig.getIndexingHashPartitionID(),

partitionConfig.getHostPositionWithinHashPartition());

Clock clock = earlybirdWireModule.provideClock();

UserUpdatesChecker userUpdatesChecker =

new UserUpdatesChecker(clock, decider, earlybirdIndexConfig.getCluster());

UserTable userTable = UserTable.newTableWithDefaultCapacityAndPredicate(

earlybirdIndexConfig.getUserTableFilter(partitionConfig)::apply);

UserScrubGeoMap userScrubGeoMap = new UserScrubGeoMap();

AudioSpaceTable audioSpaceTable = new AudioSpaceTable(clock);

SegmentSyncConfig segmentSyncConfig =

earlybirdWireModule.provideSegmentSyncConfig(earlybirdIndexConfig.getCluster());

SegmentManager segmentManager = earlybirdWireModule.provideSegmentManager(

dynamicPartitionConfig,

earlybirdIndexConfig,

earlybirdWireModule.provideSearchIndexingMetricSet(),

tweetsSearcherStats,

earlybirdServerStats,

userUpdatesChecker,

segmentSyncConfig,

userTable,

userScrubGeoMap,

clock,

criticalExceptionHandler);

QueryCacheConfig config = earlybirdWireModule.provideQueryCacheConfig(earlybirdServerStats);

QueryCacheManager queryCacheManager = earlybirdWireModule.provideQueryCacheManager(

config,

earlybirdIndexConfig,

partitionConfig.getMaxEnabledLocalSegments(),

userTable,

userScrubGeoMap,

earlybirdWireModule.provideQueryCacheUpdateTaskScheduledExecutorFactory(),

earlybirdServerStats,

tweetsSearcherStats,

decider,

criticalExceptionHandler,

clock);

EarlybirdServerSetManager serverSetManager = earlybirdWireModule.provideServerSetManager(

zkClients.discoveryClient,

dynamicPartitionConfig,

earlybirdServerStats,

EarlybirdConfig.getThriftPort(),

"");

EarlybirdWarmUpManager warmUpManager =

earlybirdWireModule.provideWarmUpManager(zkClients.discoveryClient,

dynamicPartitionConfig,

earlybirdServerStats,

decider,

clock,

EarlybirdConfig.getWarmUpThriftPort(),

"warmup\_");

EarlybirdDarkProxy earlybirdDarkProxy = earlybirdWireModule.provideEarlybirdDarkProxy(

new SearchDecider(decider),

earlybirdWireModule.provideFinagleStatsReceiver(),

serverSetManager,

warmUpManager,

partitionConfig.getClusterName());

UserUpdatesStreamIndexer userUpdatesStreamIndexer =

earlybirdWireModule.provideUserUpdatesKafkaConsumer(segmentManager);

UserScrubGeoEventStreamIndexer userScrubGeoEventStreamIndexer =

earlybirdWireModule.provideUserScrubGeoEventKafkaConsumer(segmentManager);

AudioSpaceEventsStreamIndexer audioSpaceEventsStreamIndexer =

earlybirdWireModule.provideAudioSpaceEventsStreamIndexer(audioSpaceTable, clock);

MultiSegmentTermDictionaryManager.Config termDictionaryConfig =

earlybirdWireModule.provideMultiSegmentTermDictionaryManagerConfig();

MultiSegmentTermDictionaryManager multiSegmentTermDictionaryManager =

earlybirdWireModule.provideMultiSegmentTermDictionaryManager(

termDictionaryConfig,

segmentManager,

earlybirdServerStats,

decider,

earlybirdIndexConfig.getCluster());

TermCountMonitor termCountMonitor =

earlybirdWireModule.provideTermCountMonitor(

segmentManager, earlybirdWireModule.provideTermCountMonitorScheduledExecutorFactory(),

earlybirdServerStats,

criticalExceptionHandler);

TweetCountMonitor tweetCountMonitor =

earlybirdWireModule.provideTweetCountMonitor(

segmentManager, earlybirdWireModule.provideTweetCountMonitorScheduledExecutorFactory(),

earlybirdServerStats,

criticalExceptionHandler);

ScoringModelsManager scoringModelsManager = earlybirdWireModule.provideScoringModelsManager(

earlybirdServerStats,

earlybirdIndexConfig

);

TensorflowModelsManager tensorflowModelsManager =

earlybirdWireModule.provideTensorflowModelsManager(

earlybirdServerStats,

"tf\_loader",

decider,

earlybirdIndexConfig

);

AuroraSchedulerClient schedulerClient = null;

AuroraInstanceKey auroraInstanceKey = EarlybirdConfig.getAuroraInstanceKey();

if (auroraInstanceKey != null) {

schedulerClient = new AuroraSchedulerClient(auroraInstanceKey.getCluster());

}

UpdateableEarlybirdStateManager earlybirdStateManager =

earlybirdWireModule.provideUpdateableEarlybirdStateManager(

earlybirdIndexConfig,

dynamicPartitionConfig,

zkClients.stateClient,

schedulerClient,

earlybirdWireModule.provideStateUpdateManagerExecutorFactory(),

scoringModelsManager,

tensorflowModelsManager,

earlybirdServerStats,

new SearchDecider(decider),

criticalExceptionHandler);

EarlybirdFuturePoolManager futurePoolManager = earlybirdWireModule.provideFuturePoolManager();

EarlybirdFinagleServerManager finagleServerManager =

earlybirdWireModule.provideFinagleServerManager(criticalExceptionHandler);

PartitionManager partitionManager = null;

if (EarlybirdIndexConfigUtil.isArchiveSearch()) {

partitionManager = buildArchivePartitionManager(

earlybirdWireModule,

userUpdatesStreamIndexer,

userScrubGeoEventStreamIndexer,

zkTryLockFactory,

earlybirdIndexConfig,

dynamicPartitionConfig,

segmentManager,

queryCacheManager,

earlybirdServerStats,

serverSetManager,

earlybirdWireModule.providePartitionManagerExecutorFactory(),

earlybirdWireModule.provideSimpleUserUpdateIndexerScheduledExecutorFactory(),

clock,

segmentSyncConfig,

criticalExceptionHandler);

} else {

LOG.info("Not creating PartitionManager");

}

EarlybirdSegmentFactory earlybirdSegmentFactory = new EarlybirdSegmentFactory(

earlybirdIndexConfig,

earlybirdWireModule.provideSearchIndexingMetricSet(),

tweetsSearcherStats,

clock);

EarlybirdStartup earlybirdStartup = earlybirdWireModule.provideEarlybirdStartup(

partitionManager,

userUpdatesStreamIndexer,

userScrubGeoEventStreamIndexer,

audioSpaceEventsStreamIndexer,

dynamicPartitionConfig,

criticalExceptionHandler,

segmentManager,

multiSegmentTermDictionaryManager,

queryCacheManager,

zkTryLockFactory,

serverSetManager,

clock,

segmentSyncConfig,

earlybirdSegmentFactory,

earlybirdIndexConfig.getCluster(),

searchDecider);

QualityFactor qualityFactor = earlybirdWireModule.provideQualityFactor(

decider,

earlybirdServerStats);

EarlybirdServer earlybirdServer = new EarlybirdServer(

queryCacheManager,

zkClients.stateClient,

decider,

earlybirdIndexConfig,

dynamicPartitionConfig,

partitionManager,

segmentManager,

audioSpaceTable,

termCountMonitor,

tweetCountMonitor,

earlybirdStateManager,

futurePoolManager,

finagleServerManager,

serverSetManager,

warmUpManager,

earlybirdServerStats,

tweetsSearcherStats,

scoringModelsManager,

tensorflowModelsManager,

clock,

multiSegmentTermDictionaryManager,

earlybirdDarkProxy,

segmentSyncConfig,

earlybirdWireModule.provideQueryTimeoutFactory(),

earlybirdStartup,

qualityFactor,

earlybirdWireModule.provideSearchIndexingMetricSet());

earlybirdStateManager.setEarlybirdServer(earlybirdServer);

criticalExceptionHandler.setShutdownHook(earlybirdServer::shutdown);

return earlybirdServer;

}

private PartitionManager buildArchivePartitionManager(

EarlybirdWireModule earlybirdWireModule,

UserUpdatesStreamIndexer userUpdatesStreamIndexer,

UserScrubGeoEventStreamIndexer userScrubGeoEventStreamIndexer,

ZooKeeperTryLockFactory zkTryLockFactory,

EarlybirdIndexConfig earlybirdIndexConfig,

DynamicPartitionConfig dynamicPartitionConfig,

SegmentManager segmentManager,

QueryCacheManager queryCacheManager,

SearchStatsReceiver searchStatsReceiver,

EarlybirdServerSetManager serverSetManager,

ScheduledExecutorServiceFactory partitionManagerExecutorServiceFactory,

ScheduledExecutorServiceFactory simpleUserUpdateIndexerExecutorFactory,

Clock clock,

SegmentSyncConfig segmentSyncConfig,

CriticalExceptionHandler criticalExceptionHandler)

throws IOException {

Preconditions.checkState(earlybirdIndexConfig instanceof ArchiveEarlybirdIndexConfig);

LOG.info("Creating ArchiveSearchPartitionManager");

return earlybirdWireModule.provideFullArchivePartitionManager(

zkTryLockFactory,

queryCacheManager,

segmentManager,

dynamicPartitionConfig,

userUpdatesStreamIndexer,

userScrubGeoEventStreamIndexer,

searchStatsReceiver,

(ArchiveEarlybirdIndexConfig) earlybirdIndexConfig,

serverSetManager,

partitionManagerExecutorServiceFactory,

simpleUserUpdateIndexerExecutorFactory,

earlybirdWireModule.provideSearchIndexingMetricSet(),

clock,

segmentSyncConfig,

criticalExceptionHandler);

}

}