package com.twitter.search.earlybird.config;

import java.util.Date;

import com.google.common.base.Preconditions;

/\*\*

\* A simple wrapper around TierInfo that returns the "real" or the "overriden" values from the given

\* {@code TierInfo} instance, based on the given {@code useOverrideTierConfig} flag.

\*/

public class TierInfoWrapper implements ServingRange {

private final TierInfo tierInfo;

private final boolean useOverrideTierConfig;

public TierInfoWrapper(TierInfo tierInfo, boolean useOverrideTierConfig) {

this.tierInfo = Preconditions.checkNotNull(tierInfo);

this.useOverrideTierConfig = useOverrideTierConfig;

}

public String getTierName() {

return tierInfo.getTierName();

}

public Date getDataStartDate() {

return tierInfo.getDataStartDate();

}

public Date getDataEndDate() {

return tierInfo.getDataEndDate();

}

public int getNumPartitions() {

return tierInfo.getNumPartitions();

}

public int getMaxTimeslices() {

return tierInfo.getMaxTimeslices();

}

public TierConfig.ConfigSource getSource() {

return tierInfo.getSource();

}

public boolean isEnabled() {

return tierInfo.isEnabled();

}

public boolean isDarkRead() {

return getReadType() == TierInfo.RequestReadType.DARK;

}

public TierInfo.RequestReadType getReadType() {

return useOverrideTierConfig ? tierInfo.getReadTypeOverride() : tierInfo.getReadType();

}

public long getServingRangeSinceId() {

return useOverrideTierConfig

? tierInfo.getServingRangeOverrideSinceId()

: tierInfo.getServingRangeSinceId();

}

public long getServingRangeMaxId() {

return useOverrideTierConfig

? tierInfo.getServingRangeOverrideMaxId()

: tierInfo.getServingRangeMaxId();

}

public long getServingRangeSinceTimeSecondsFromEpoch() {

return useOverrideTierConfig

? tierInfo.getServingRangeOverrideSinceTimeSecondsFromEpoch()

: tierInfo.getServingRangeSinceTimeSecondsFromEpoch();

}

public long getServingRangeUntilTimeSecondsFromEpoch() {

return useOverrideTierConfig

? tierInfo.getServingRangeOverrideUntilTimeSecondsFromEpoch()

: tierInfo.getServingRangeUntilTimeSecondsFromEpoch();

}

public static boolean servingRangesOverlap(TierInfoWrapper tier1, TierInfoWrapper tier2) {

return (tier1.getServingRangeMaxId() > tier2.getServingRangeSinceId())

&& (tier2.getServingRangeMaxId() > tier1.getServingRangeSinceId());

}

public static boolean servingRangesHaveGap(TierInfoWrapper tier1, TierInfoWrapper tier2) {

return (tier1.getServingRangeMaxId() < tier2.getServingRangeSinceId())

|| (tier2.getServingRangeMaxId() < tier1.getServingRangeSinceId());

}

}