

Introducing redis as a distributed cache for queries

Usage Instructions

if you want to use segment cache, I suggest you to use Memcached cache

1. Single Redis Node or proxy mode

1) modify `applicationContext.xml`

```
<bean id="jedisPoolConfig" class="redis.clients.jedis.JedisPoolConfig">
    <property name="maxTotal" value="30"/>
    <property name="maxIdle" value="10"/>
    <property name="numTestsPerEvictionRun" value="1024"/>
    <property name="timeBetweenEvictionRunsMillis" value="30000"/>
    <property name="minEvictableIdleTimeMillis" value="1800000"/>
    <property name="softMinEvictableIdleTimeMillis" value="10000"/>
    <property name="maxWaitMillis" value="1500"/>
    <property name="testOnBorrow" value="true"/>
    <property name="testWhileIdle" value="true"/>
    <property name="blockWhenExhausted" value="false"/>
</bean>

<bean id="connectionFactory"

    class="org.springframework.data.redis.connection.jedis.JedisConnectionFactory"
    >
    <property name="poolConfig" ref="jedisPoolConfig"/>
    <property name="port" value="${kylin.cache.redis.port}"/>
    <property name="hostName" value="${kylin.cache.redis.host}"/>
    <property name="password" value="${kylin.cache.redis.passwd}"/>
    <property name="timeout" value="${kylin.cache.redis.timeout}"/>
</bean>

<bean id="redisTemplate"
    class="org.springframework.data.redis.core.RedisTemplate">
    <property name="connectionFactory" ref="connectionFactory"/>
    <property name="keySerializer">
```

```

        <bean
class="org.springframework.data.redis.serializer.GenericJackson2JsonRedisSeria
lizer"/>
        </property>
        <property name="valueSerializer">
        <bean
class="org.springframework.data.redis.serializer.GenericJackson2JsonRedisSeria
lizer"/>
        </property>
</bean>
<bean id="remoteCacheManager"
class="org.apache.kylin.cache.cachemanager.RedisCacheManagerAdaptor">
    <constructor-arg name="redisOperations" ref="redisTemplate"/>
    <constructor-arg name="cacheNames">
        <list>
            <value>StorageCache</value>
            <value>ExceptionQueryCache</value>
            <value>UserCache</value>
        </list>
    </constructor-arg>
    <property name="expires" >
        <map key-type="java.lang.String" value-type="java.lang.Long">
            <entry key="StorageCache" value="86400"/>
            <entry key="ExceptionQueryCache" value="86400"/>
            <entry key="UserCache" value="10800"/>
        </map>
    </property>
</bean>

```

2. add related config in `kylin.properties`

```

kylin.cache.redis.port=7777
kylin.cache.redis.host=127.0.0.1
kylin.cache.redis.passwd=123456
kylin.cache.redis.timeout=5000

```

2. redis cluster mode

1) modeify `applicationContext.xml`

```

<bean id="ehcache"
    class="org.springframework.cache.ehcache.EhCacheManagerFactoryBean"
    p:configLocation="classpath:ehcache-test.xml" p:shared="true"/>
<bean id="localCacheManager"
class="org.apache.kylin.cache.cachemanager.InstrumentedEhCacheCacheManager"
    p:cacheManager-ref="ehcache"/>

<bean id="jedisPoolConfig" class="redis.clients.jedis.JedisPoolConfig">

```

```

    <property name="maxTotal" value="30" />
    <property name="maxIdle" value="10" />
    <property name="numTestsPerEvictionRun" value="1024" />
    <property name="timeBetweenEvictionRunsMillis" value="30000" />
    <property name="minEvictableIdleTimeMillis" value="1800000" />
    <property name="softMinEvictableIdleTimeMillis" value="10000" />
    <property name="maxWaitMillis" value="1500" />
    <property name="testOnBorrow" value="true" />
    <property name="testWhileIdle" value="true" />
    <property name="blockWhenExhausted" value="false" />
</bean>
<bean id="clusterConfiguration"
class="org.springframework.data.redis.connection.RedisClusterConfiguration">
    <constructor-arg name="clusterNodes">
        <list>
            <value>127.0.0.1:7777</value>
            <value>127.0.0.1:7778</value>
            <value>127.0.0.1:7779</value>
        </list>
    </constructor-arg>
</bean>

<bean id="connectionFactory"

    class="org.springframework.data.redis.connection.jedis.JedisConnectionFactory"
">
    <constructor-arg name="clusterConfig" ref="clusterConfiguration" />
    <constructor-arg name="poolConfig" ref="jedisPoolConfig" />
</bean>

<bean id="redisTemplate"
class="org.springframework.data.redis.core.RedisTemplate">
    <property name="connectionFactory" ref="connectionFactory" />
    <property name="keySerializer">
        <bean
class="org.springframework.data.redis.serializer.GenericJackson2JsonRedisSeriali
lizer" />
    </property>
    <property name="valueSerializer">
        <bean
class="org.springframework.data.redis.serializer.GenericJackson2JsonRedisSeriali
lizer" />
    </property>
</bean>
<bean id="remoteCacheManager"
class="org.apache.kylin.cache.cachemanager.RedisCacheManagerAdaptor">
    <constructor-arg name="redisOperations" ref="redisTemplate" />
    <constructor-arg name="cacheNames">
        <list>

```

```

        <value>StorageCache</value>
        <value>ExceptionQueryCache</value>
        <value>UserCache</value>
    </list>
</constructor-arg>
<property name="expires" >
    <map key-type="java.lang.String" value-type="java.lang.Long">
        <entry key="StorageCache" value="86400"/>
        <entry key="ExceptionQueryCache" value="86400"/>
        <entry key="UserCache" value="10800"/>
    </map>
</property>
</bean>
<bean id="cacheManager"
class="org.apache.kylin.cache.cachemanager.RemoteLocalFailOverCacheManager"/>

```

Need to change `clusterNodes` values to your cluster nodes ip