

Assignment4 - code review

1. Populate database 的时候

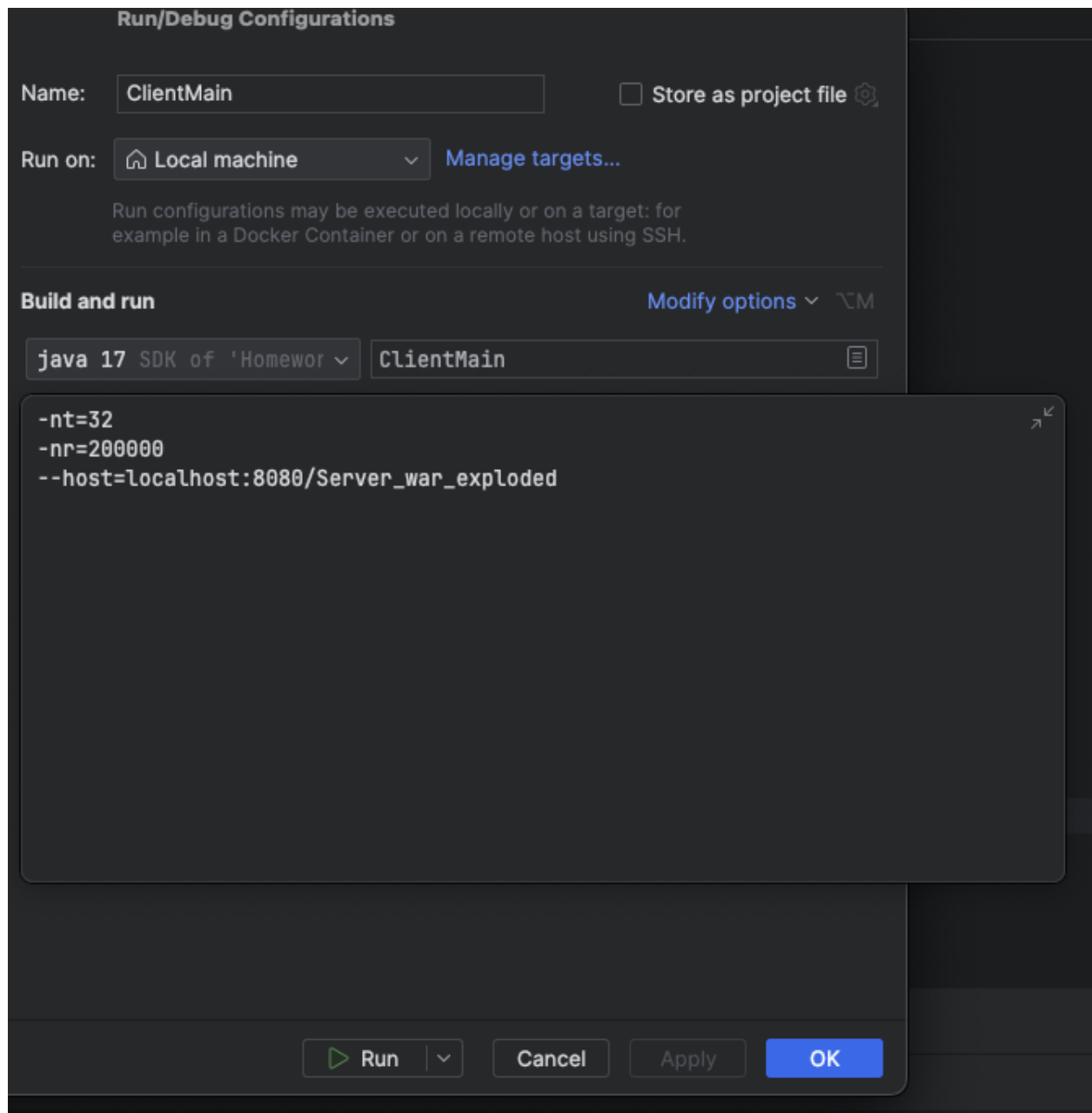
- keep resort and season constant in all 3 runs
- vary the day value for each run, namely {1, 2, 3}
- keep the number of skiers, lift rides as default values for first 3 assignments
- Thread configuration of your choice

resort 写 1

season写 2024

day {1,2,3}

2. 如果我没理解错的话，client server的CLI是这么用的吧？



3. Redis

In order to satisfy the requests, I used skierID as primary key, so that we can easily retrieve the details for skier N.

The value for each key consists of: `dayID="_".join(days)`, `liftID="_".join(lift)` and `resortID="_".join(resorts)`.

I also used `"day{$dayID$}"` as another primary key, and the value consists of resortID and skierID.

```

1 usage  ± Xunyan Zhang
private void writeToDB(String entry) {
    try (Jedis jedis = jedisPool.getResource()) {
        JSONObject json = gson.fromJson(entry, JSONObject.class);
        String skierId = String.valueOf(json.get("skierID"));
        String time = String.valueOf(json.get("time"));
        String liftId = String.valueOf(json.get("liftID"));
        String resortId = String.valueOf(json.get("resortID"));
        String dayId = String.valueOf(json.get("dayID"));
        Map<String, String> storedEntry = jedis.hgetAll(skierId);
        if (storedEntry.isEmpty()) {
            storedEntry = new HashMap<> (storedEntry);
        }
        storedEntry.put("time", storedEntry.getOrDefault( key: "time", defaultValue: "") + DB_SEPARATOR + time);
        storedEntry.put("liftID", storedEntry.getOrDefault( key: "liftID", defaultValue: "") + DB_SEPARATOR + liftId);
        storedEntry.put("resortID", storedEntry.getOrDefault( key: "resortID", defaultValue: "") + DB_SEPARATOR + resortId);
        storedEntry.put("dayID", storedEntry.getOrDefault( key: "dayID", defaultValue: "") + DB_SEPARATOR + dayId);
        jedis.hmset(skierId, storedEntry);
    } catch (Exception e) {
        Consumer.logger.error(Arrays.toString(e.getStackTrace()));
    }
}
}

```

你代码这里没有传seasonID吧

```

127.0.0.1:6379> hgetAll 456
1) "liftID"
2) "_1_1"
3) "resortID"
4) "_123_116"
5) "dayID"
6) "_100_201"
7) "time"
8) "_360_360"
127.0.0.1:6379>

```

你想的是再创建一个以dayID为key，然后里面是这样的结构？

1. "ResortID"
2. "_1_1"
3. "skierID"
4. "_123_456"

这样的结构，我没想到什么好方法可以快速取到the number of unique skiers at given season, resort, day.

```
redis-cli --raw hget "dayN" | awk '{print $1}' | grep '^X_' | cut -d'_' -f2 | sort -u | wc -l
```

这个指令并不能查到

除非是在服务器端逻辑来拆分字符串然后对unique number进行计算（这可能要写道算法题

```
public int getUniqueSkierCount(String dayID, String resortID,
String seasonID) {
    // Get all data associated with dayID from Redis
    Map<String, String> storedEntry = jedis.hgetAll(dayID);

    // Get the resortID, seasonID, and skierID data stored in
the hash
    String resortIDs = storedEntry.get("resortID");
    String seasonIDs = storedEntry.get("seasonID");
    String skierIDs = storedEntry.get("skierID");

    // Check if there are matching resortID and seasonID
    if (resortIDs != null && seasonIDs != null && skierIDs !=
null) {
        // Split the strings into arrays
        String[] resortArray = resortIDs.split("_");
```

```

String[] seasonArray = seasonIDs.split("_");
String[] skierArray = skierIDs.split("_");

// Create a set to store unique skierID
Set<String> uniqueSkiers = new HashSet<>();

// Traverse the arrays, checking for matching resortID and seasonID
for (int i = 0; i < skierArray.length; i++) {
    if (resortArray[i].equals(resortID) && seasonArray[i].equals(seasonID)) {
        uniqueSkiers.add(skierArray[i]);
    }
}

// Return the number of unique skierID
return uniqueSkiers.size();
}

// If no matching data is found, return 0
return 0;
}

```

但是这种办法势必造就查询的速度过慢，设计更复杂的redis键值对数据那就对写的操作会有影响

我提供一种设计方法：

为每个 `dayID`，`resortID` 和 `seasonID` 组合存储唯一 `skier` ID 的集合

- 使用 `SET` 数据结构，键可以是 `"day:<dayID>:resort:<resortID>:season:<seasonID>:skiers"`

// 添加记录时的命令

```
String key = String.format("day:%s:resort:%s:season:%s:skiers", dayId, resortId,
```

```
seasonId);  
jedis.sadd(key, String.valueOf(skierId));  
//查询  
long uniqueSkierCount = jedis.scard(key);  
System.out.println("Unique skiers count: " + uniqueSkierCount);
```