

牛人计划-中级项目课（9）



牛客网
NOWCODER





课程目录

CONTENTS

排序

- 通用排序
- Hacker News
- Reddit
- StackOverflow
- IMDB

多线程

- 简介
- Thread , Synchronized
- BlockingQueue , AtomicInteger
- ThreadLocal , Executor , Future



通用排序

1. 单位时间内的交互数，del.icio.us按1小时内收藏排行
2. 总交互数，按总点赞数
3. 评论数加权
4. 按时间排序



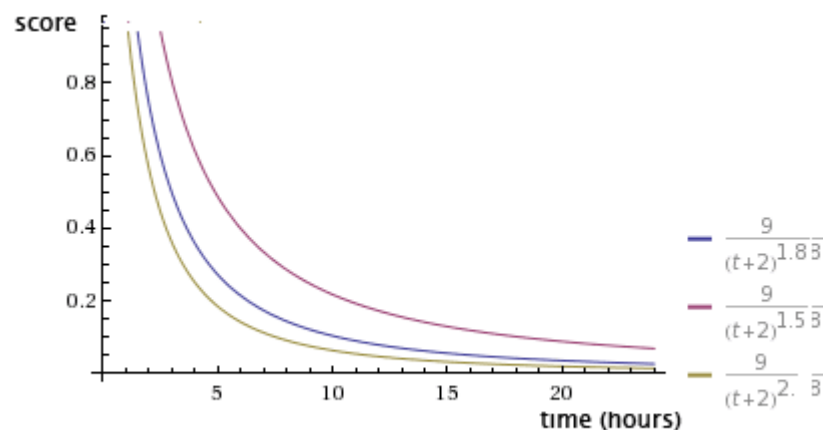
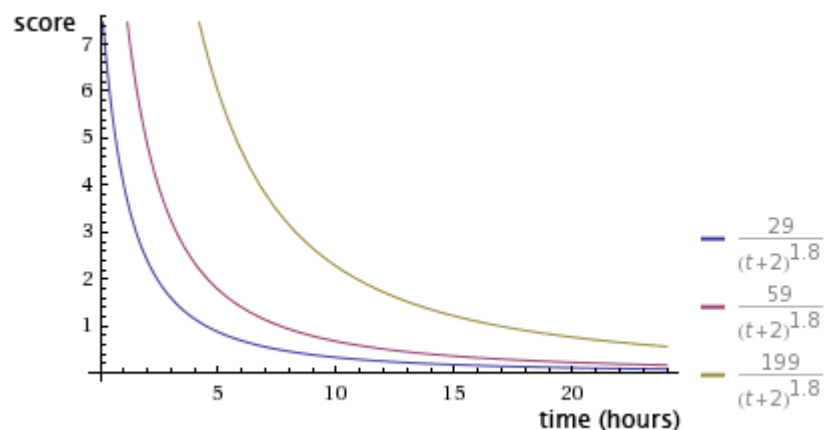
Hacker News <https://news.ycombinator.com/>

$$\text{Score} = (P - 1) / (T + 2)^G$$

P：投票数，-1是把自己投的过滤掉

T：发布到现在的时间间隔，单位小时，+2防止除数太小

G：重力加速度，分值根据时间降低速率



<https://medium.com/hacking-and-gonzo/how-hacker-news-ranking-algorithm-works-1d9b0cf2c08d>



Reddit <https://www.reddit.com/>

Given the time the entry was posted A and the time of 7:46:43 a.m. December 8, 2005 B , we have t_s as their difference in seconds

$$t_s = A - B$$

and x as the difference between the number of up votes U and the number of down votes D

$$x = U - D$$

where $y \in \{-1, 0, 1\}$

$$y = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}$$

and z as the maximal value, of the absolute value of x and 1

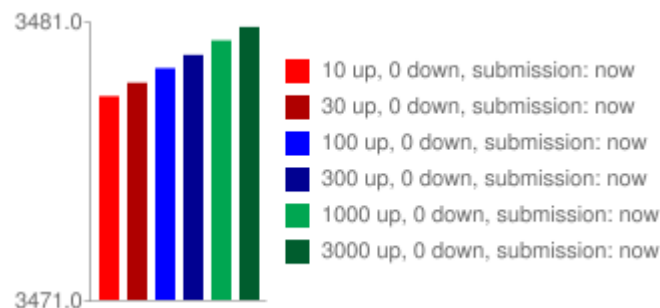
$$z = \begin{cases} |x| & \text{if } |x| \geq 1 \\ 1 & \text{if } |x| < 1 \end{cases}$$

we have the rating as a function $f(t_s, y, z)$

$$f(t_s, y, z) = \log_{10} z + \frac{y t_s}{45000}$$

$86400/45000=1.92$ 一天权重调整
 $10^{1.92}=83$ 投票差要涨83倍

赞成票加分，投票差前10和接下来的100等权



时间是最重要的权重，由于流量比较大，所以对于高赞文章有所优势，适合新闻类排序

<https://medium.com/hacking-and-gonzo/how-reddit-ranking-algorithms-work-ef111e33d0d9>



StackOverflow <http://stackoverflow.com>

$$\frac{(\log(Qviews)*4) + ((Qanswers * Qscore)/5) + \text{sum}(Ascores)}{((QageInHours+1) - ((QageInHours - Qupdated)/2)) ^ 1.5}$$

Qviews : 问题浏览数, 通过log来平滑

Qanswer : 问题回答数, 有回答的题目才是好问题

Qscore : 问题赞踩差, 赞的越多, 问题越好

sum (Ascores) : 回答赞踩差, 回答的越多问题越好

QageInHours : 题目发布时间差, 时间越久排名越后

Qupdated : 最新的回答时间, 越新关注度越高

<http://meta.stackexchange.com/questions/11602/what-formula-should-be-used-to-determine-hot-questions>



IMDB <http://www.imdb.com/chart/top>

加权排名 (WR) = $(v \div (v+m)) \times R + (m \div (v+m)) \times C$

R = 某电影投票平均分

v = 有效投票人数

m = 最低投票人数, 1250

C = 所有电影平均值

投票人数越多, 越偏向于用户打分值, 防止冷门电影小数人高分导致的高分

http://www.imdb.com/help/show_leaf?votestopfaq

<https://community.topcoder.com/longcontest/?module=Static&d1=support&d2=ratings>



多线程简介

优势

- 充分利用多处理器
- 可以异步处理任务



挑战

- 数据会被多个线程访问，有安全性问题
- 不活跃的线程也会占用内存资源
- 死锁



Thread

1. extends Thread , 重载run()方法
2. implements Runnable() , 实现run()方法

```
new Thread(new Runnable() {  
    @Override  
    public void run() {  
        Random random = new Random();  
        for (int i = 0; i < 10; ++i) {  
            sleep(random.nextInt(1000));  
            System.out.println(String.format("T%d : %d", tid, i));  
        }  
    }  
}, String.valueOf(i)).start();
```



Synchronized - 内置锁

1. 放在方法上会锁住所有synchronized方法
2. synchronized(obj) 锁住相关的代码段

```
public static void testSynchronized1() {  
    synchronized (obj) {  
        Random random = new Random();  
        for (int i = 0; i < 10; ++i) {  
            sleep(random.nextInt(1000));  
        }  
    }  
}
```



BlockingQueue 同步队列

Summary of BlockingQueue methods

	<i>Throws exception</i>	<i>Special value</i>	<i>Blocks</i>	<i>Times out</i>
Insert	<u>add(e)</u>	<u>offer(e)</u>	<u>put(e)</u>	<u>offer(e, time, unit)</u>
Remove	<u>remove()</u>	<u>poll()</u>	<u>take()</u>	<u>poll(time, unit)</u>
Examine	<u>element()</u>	<u>peek()</u>	<i>not applicable</i>	<i>not applicable</i>



ThreadLocal

1. 线程局部变量。即使是一个static成员，每个线程访问的变量是不同的。
2. 常见于web中存储当前用户到一个静态工具类中，在线程的任何地方都可以访问到当前线程的用户。
3. 参考HostHolder.java里的users



Executor

Executor

1. 提供一个运行任务的框架。
2. 将任务和如何运行任务解耦。
3. 常用于提供线程池或定时任务服务

```
ExecutorService service = Executors.newFixedThreadPool(2);
service.submit(new Runnable() {
    @Override
    public void run() {
        for (int i = 0; i < 10; ++i) {
            sleep(1000);
            System.out.println("Execute %d" + i);
        }
    }
});
```



Future

```
public static void testFuture() {
    ExecutorService service = Executors.newSingleThreadExecutor();
    Future<Integer> future = service.submit(new Callable<Integer>() {
        @Override
        public Integer call() throws Exception {
            sleep(1000);
            //throw new IllegalArgumentException("一个异常");
            return 1;
        }
    });
    service.shutdown();

    try {
        System.out.println(future.get());
        //System.out.println(future.get(100, TimeUnit.MILLISECONDS));
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

1. 返回异步结果
2. 阻塞等待返回结果
3. timeout
4. 获取线程中的Exception



Thanks



牛客网
NOWCODER