Questions | Jobs | Documentation Beta | Tags | Users | Badges | Ask Question
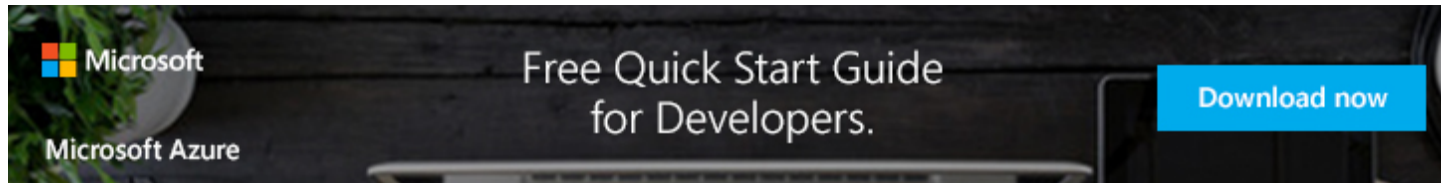
LFU cache java implementation

asked   6 months ago

viewed   167 times

0

LFU cache using a priority queue: Using a HashMap for fast lookup and a priority queue to get rid of lowest priority Item:

```
public class LFUCacheWithPriorityQ<K, V> {

Map<K, Node> myMap = new HashMap<>();
PriorityQueue<Node> priorityQueue = new PriorityQueue<>();
int count = 0;

public void add(K key, V value) {
  LFUCacheWithPriorityQ<K, V>.Node node = myMap.get(key);
  if (node != null) {
    node.v = value;
    return;
  }

  while (count > 10) {
    // remove old entry
    LFUCacheWithPriorityQ<K, V>.Node remove = priorityQueue.remove();
    myMap.remove(remove.k);
    count--;
  }

  LFUCacheWithPriorityQ<K, V>.Node newNode = new LFUCacheWithPriorityQ.Node();
  newNode.k = key;
  newNode.v = value;
  newNode.count = 1;
```

Jobs near you

Software Development Engineer, Web Framework

Audible   ⚲ *Newark, NJ*

🚚 **RELOCATION**

http   hls

Senior Software Development Engineer

Audible   ⚲ *Newark, NJ*

```
    myMap.put(key, newNode);

    priorityQueue.add(newNode);
    count++;
}

public V get(K key) {
    LFUCacheWithPriorityQ<K, V>.Node node = myMap.get(key);
    if (node == null) {
        return null;
    }
    priorityQueue.remove(node);
    node.count = node.count + 1;
    priorityQueue.add(node);
```

As you see the issue is "priorityQueue.remove(node);" which is linear time, I wanted to know what I need to do to make it at least log N.

Here is another implementation using a TreeMap, here the runtime is

```
public class LFUCacheWithOnlySortedMap<K, V> {

class MyKey implements Comparable<MyKey> {
    K key;
    int count;

    @Override
    public int compareTo(LFUCacheWithOnlySortedMap<K, V>.MyKey o) {
        return this.count - o.count;
    }

    @Override
    public boolean equals(Object o) {
        return key.equals(o);
    }
}

class MyValue {
    public MyValue(V value, int i) {
        this.value = value;
        this.count = i;
    }

    V value;
    int count;
}
```

Related

```java
int count = 0;
SortedMap<MyKey, MyValue> sortedMap = new TreeMap<>();

public void add(K key, V value) {
    MyKey myKey = new MyKey();
    myKey.key = key;
    MyValue oldValue = sortedMap.get(myKey);
    if (oldValue != null) {
        oldValue.value = value;
        return;
    }
}
```

}

`java` `caching`

share improve this question

edited Jun 21 at 22:22

asked Jun 21 at 7:08

andy
11 3

You would need to use a different data structure which is also log N for remove. This could be achieved if you had a data structure which maintained two indexes to the node. BTW I would use LinkedHashMap as an LRU cache as this would be much simpler. The downside of the current LFU is that once you have entries with a high enough count, no new entry has much chance of replacing it. In which case you may as well just use a fixed size map if the values with the highest count and not cache any new values. – Peter Lawrey Jun 21 at 7:36

I understand that a LRU is better and can be implemented by programcreek.com/2013/03/leetcode-lru-cache-java. I am trying to implement LFU for my own understanding. – andy Jun 21 at 22:19

A LFU would need to be consider the time period for the frequency part. If you do not, you will get to the point where no new entry will get as much count as those in the cache. I suspect there is no way to make a LFU cache practical. – Peter Lawrey Jun 22 at 10:26

add a comment

Know someone who can answer? Share a link to this question via email, Google+, Twitter, or Facebook.

Your Answer

Hot Network Questions

- What does my girlfriend want me to buy for her?
- 6 Water Glasses Upside Down
- What is the day to day life like as a father?
- How do you communicate with antimatter beings?
- How can I tell whether a generator was just-started?
- Simple pack Uri builder
- Is an ACK necessary when using reliable protocols like TCP? How much overhead / throughput penalty does it create?
- Is there an equation for every graph?
- Shortest auto-destructive loop
- Does my mother's car have a thermostat problem?
- How to stop cups sticking to placemats
- A published paper stole my unpublished results from a science fair
- Why did it take longer to go to Rivendell in The Hobbit than in The Fellowship of the Ring?
- Is the form "double Dutch" still used?
- How can two laptops have the same resolution and screen size but different pixel densities?
- How to choose origin in rotational problems to calculate torque?

B **B** **B** **B** **B** **B** **B** **B** **B** **B** **B** **B**

Browse other questions tagged java caching or ask your own question.

Word for fake religious people

What is the latest version of Ubuntu 14.04 on which fglrx for AMD can be installed?

Who were the red-robed citizens of Jedha City?

Colored cells in a table and multicol

Hide the clock on the iPhone 6+ lockscreen

Crazy 8s Code Golf

What is the truth about 1.5V "lithium" cells

Is it bad form to write mysterious proofs without explaining what one intends to do?

question feed

TECHNOLOGY

Stack Overflow

Server Fault

Super User

Web Applications

Ask Ubuntu

Webmasters

Game Development

TeX - LaTeX

Software Engineering

Unix & Linux

Ask Different (Apple)

WordPress Development

Geographic Information Systems

Electrical Engineering

Android Enthusiasts

Information Security

Database Administrators

Drupal Answers

SharePoint

User Experience

Mathematica

Salesforce

ExpressionEngine® Answers

Cryptography

Code Review

Magento

Signal Processing

Raspberry Pi

Programming Puzzles & Code Golf

more (7)

LIFE / ARTS

Photography

Science Fiction & Fantasy

Graphic Design

Movies & TV

Music: Practice & Theory

Seasoned Advice (cooking)

Home Improvement

Personal Finance & Money

Academia

more (8)

CULTURE / RECREATION

English Language & Usage

Skeptics

Mi Yodeya (Judaism)

Travel

Christianity

English Language Learners

Japanese Language

Arqade (gaming)

Bicycles

Role-playing Games

Anime & Manga

Motor Vehicle Maintenance & Repair

more (17)

SCIENCE

MathOverflow

Mathematics

Cross Validated (stats)

Theoretical Computer Science

Physics

Chemistry

Biology

Computer Science

Philosophy

more (3)

OTHER

Meta Stack Exchange

Stack Apps

Area 51

Stack Overflow Talent