< Back | Python3 Multithreaded BFS with Queue

kuznecpl ★ 75 Last Edit: November 1, 2019 5:18 PM 900 VIEWS

15

I use python3's multithreaded Queue to implement BFS traversal of urls.
A lock is acquired before adding a new url to the queue to ensure that no url is put twice. (imagine two threads trying to add a new url at the same time).

```
import threading
from queue import Queue

class Solution:
    def crawl(self, startUrl: str, htmlParser: 'HtmlParser') -> List[str]:
        def hostname(url):
            start = len("http://")
            i = start
            while i < len(url) and url[i] != "/":
                i += 1
            return url[start:i]

        queue = Queue()
        seen = {startUrl}
        start_hostname = hostname(startUrl)
        seen_lock = threading.Lock()

        def worker():
            while True:
                url = queue.get()
                if url is None:
                    return

                for next_url in htmlParser.getUrls(url):
                    if next_url not in seen and hostname(next_url) == start_hostname:
                        seen_lock.acquire()
                        # Acquire lock to ensure urls are no enqueed multiple times
                        if next_url not in seen:
                            seen.add(next_url)
                            queue.put(next_url)
                        seen_lock.release()
                queue.task_done()

        num_workers = 8
        workers = []
        queue.put(startUrl)

        for i in range(num_workers):
            t = threading.Thread(target=worker)
            t.start()
            workers.append(t)

        # Wait until empty
        queue.join()

        for i in range(num_workers):
            queue.put(None)
        for t in workers:
            t.join()

        return list(seen)
```

Comments: 1 Best | Most Votes | Newest to Oldest | Oldest to Newest

Type comment here... (Markdown is supported)

Post

ping999 ★ 36 May 3, 2020 9:23 AM

best python solution!

0 Reply