分布式系统第一次作业

q1文档部分

1752854

赵羿昕

# 代码环境：

Python3.7

# 使用库：

import time                         #计时

import mmap                         #方便读dat文件

import os                           #获取路径

import numpy as np                  #用于画图

import matplotlib.pyplot as plt     #用于画图

from scipy.interpolate import spline#用于拟合曲线

# 核心函数：

## 从sample中提取单词并去重排序

def SelectWords(path):

    f = open(path,'r')

    WordList = []

    for i in f:

        WordList +=(re.split('[,/.\\\n \t\r:|]',i))

    f.close()

    WordList = list(set(WordList))

    WordList.sort(key=custom\_key)

    while WordList[0] == '':

        del WordList[0]

    return WordList

## 2.将单词转化为二进制按kb保存进sort.dat中

def BytesWrite(WordList, pathDat):

    WordListB = []

    for i in WordList:

        length = str(len(i))

        word = length

        word += i

        WordListB.append(word.encode("utf-8"))

    KBlist = WordListB[0]

    f = open(pathDat, 'w+')

    for i in WordListB[1:]:

        if len(KBlist) + len(i) > 1024:

            print(KBlist, file = f, end = '')

            KBlist = i

        else:

            KBlist += i

    if len(KBlist) > 1:

        print(KBlist, file = f, end = '')

    f.close()

## 3.在sort中查找test单词

def SearchWords():

    t = open(pathTxt, 'r')

    l = open(pathLog, 'a+')

    fd = os.open(pathDat, os.O\_RDWR)

    m = mmap.mmap(fd, 0,  access= mmap.ACCESS\_READ)

    wList = []

    for i in t:

        wList += i.split()

    for i in range(0, len(wList)):

        findNo = m.find(wList[i].encode("utf-8"))

        if findNo == -1:

            print("notfound", file= l)

            continue

        findLength = len(wList[i])

        site = findNo + findLength

        rstLength = ''

        while chr(m[site]) in '0123456789':

            rstLength += str(chr(m[site]))

            site += 1

        print(m[site:(site+int(rstLength))].decode(), file= l)

    t.close()

    l.close()

## 4.绘图保存并展示

def draw():

    f = open(pathLog, 'r')

    data = []

    drawdata = []

    for i in f:

        try:

            b = float(i)

            data.append(b)

        except:

            continue

    i = 0

    if len(data) < 19:

        return

    while i < len(data):

        drawdata.append(data[i]/data[i+1])

        i += 2

    x = [j for j in range(1, len(drawdata)+1)]

    y = drawdata

    xnew = np.linspace(1,len(drawdata),300)

    y\_smooth = spline(x,y,xnew)

    plt.figure(figsize=(8,4))

    plt.plot(xnew,y\_smooth, label="$sin(x)$",color="red",linewidth=2)

    plt.xlabel("Times")

    plt.ylabel("time")

    plt.title("store time / search time")

    plt.savefig(module\_path + "out\\result.png")

    plt.show()

# 执行结果（用自己的数据绘的图，共12次，波动有数据量的原因也有电脑的原因，根据新数据生成的图片在out文件夹下）：

