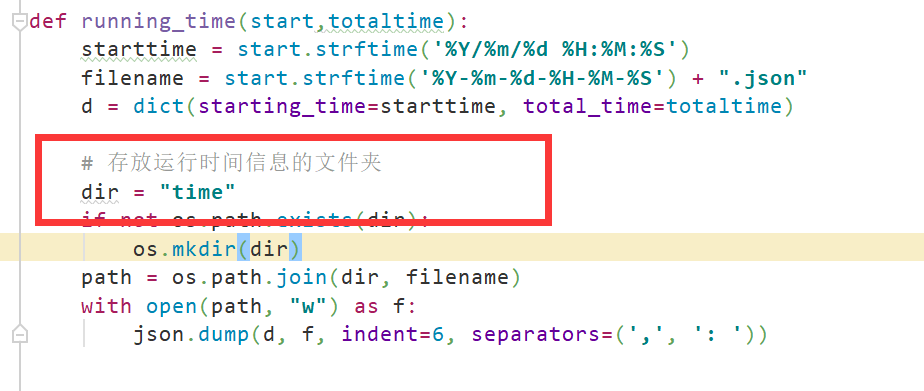
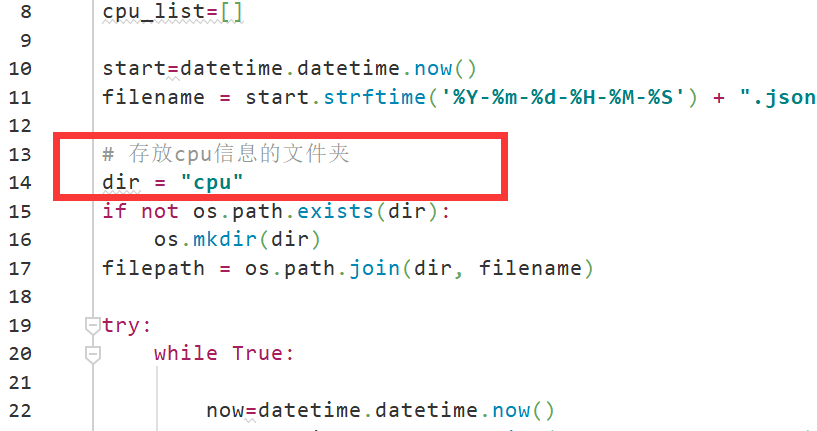
1. 看一下client和server的端口号是不是一样的，不一样的话改成一样的，我测试的时候改了一些端口号。
2. 这个文件夹比fault-tolerance-version1文件夹多了一个cpu-percent.py文件，这个文件是用来记录CPU的使用率的，因为我不太清楚把记录CPU使用率的代码嵌入到client中并单独开一个线程对于不嵌入会有多大的影响，所以我在这个文件夹中把记录CPU使用率的代码单独做了一个.py文件。
3. client程序每次运行会生成1个.json文件，且每个文件是以时间为文件名的。该文件是用来记录运行总时间的，文件是放在自己创建的文件夹中的，文件夹的名称在如下代码处修改：



文件的名称改成“结构名称(bcdc/fattree/DCell)-实验名称(fault-tolerance)-记录信息的类型(cpu/time)”，如”Dcell-fault-tolerance -time”。

4．cpu-percent.py每次运行会生成1个.json文件，且每个文件是以时间为文件名的。该文件是用来记录运行CPU使用率的，文件是放在自己创建的文件夹中的，文件夹的名称在如下代码处修改：



文件名取法同上。

1. 其他部署和之前一样。

6． 两个程序都运行在linux终端上，不要运行在pycharm这种IDE上，因为我的cpu-percent.py程序是针对在终端上运行来写的。运行时，先运行cpu-percent.py程序，紧接着运行client程序。当client结束时，用control+C来结束cpu-percent.py的运行，这个要手动结束，因为cpu-percent.py这个程序是个死循环。cpu-percent.py运行结束之后就可以了。

7．实验运行3次，每次运行完之后，查看一下两个.json文件，看看有没有异常值，比如值特别大的，或者所有的值都是一样的。有异常值的话再运行一次。有异常值的文件不用删掉，保留即可。

8．运行完成之后，将上面两个包含信息的文件夹发给我。