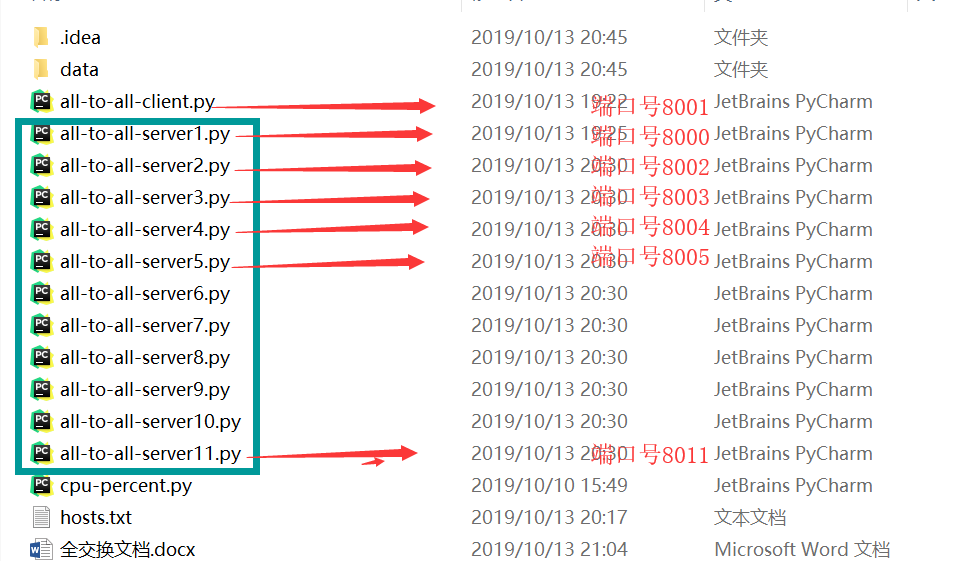
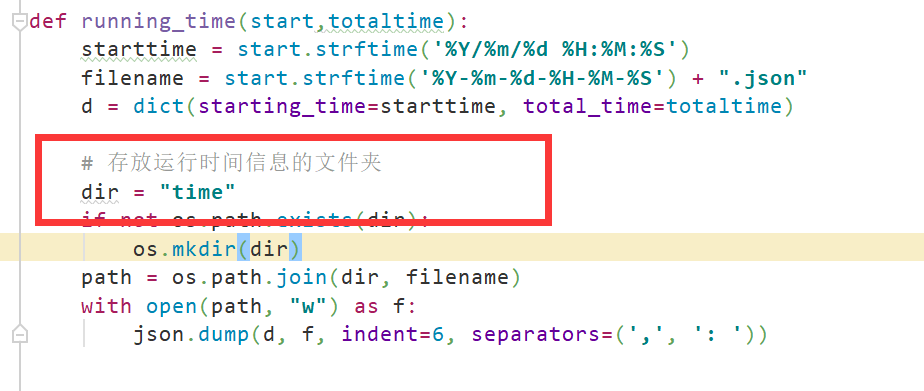
1. 全交换通信是12台机器同时向除本机外的另外11台机器传文件，广播是1对11，全交换是12个1对11，而且是同时。
2. 代码和广播时的代码是一样的，只是麻烦了一点，需要同时开12个终端，其中1个终端部署client程序，另外11个终端部署server程序。先运行11个终端server程序，再运行那个client终端程序，如下所示。

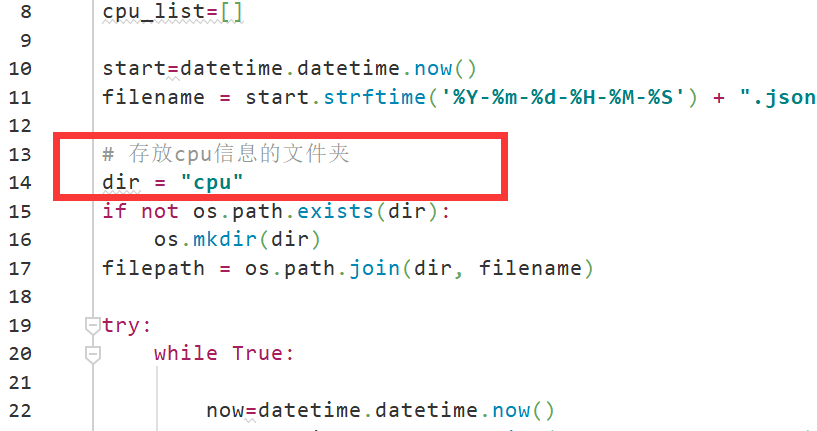


1. hosts.txt中存放除本机外的11个ip地址。
2. 这里最需要注意的是端口号的问题，将12台主机上client的端口号依次改为8000到8011，每台机器上的另外11个server程序的端口号依次改为除client中的端口号的其他所有端口号。例如，第一台机器中client的端口号为8000，则另外11个server上的端口号依次为8001到8011。第二台机器的client的端口号为8001，则另外11个server上的端口号依次为8000,8002到8011，如上图所示。
3. client程序每次运行会生成1个.json文件，且每个文件是以时间为文件名的。该文件是用来记录运行总时间的，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(all-to-all)-记录信息的类型(cpu/time)”，如”Dcell-all-to-all -time”。

1. 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。

7. 其他部署和之前一样。

8． 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。当client程序结束时，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。

9．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。

10．运行完成之后，将上面两个包含信息的文件夹发给我。