

主要使用main.py跑，里面是核心部分。

主要功能：

1. 存放组网信息以供其他模块使用
2. 维护当前周期内的各链路上的在线流量
3. 维护当前周期内的各链路的实际剩余流量
4. 维护当前周期内的各链路的预测剩余流量
5. 载入 task.txt，初始化调度任务
6. 初始化用于存放每轮调度结果的 txt 文件
7. 每轮周期结束后，将各链路的利用率记录到 txt 文件中（output\_文件夹里面）：

计算各链路的总利用率（在线流量加离线流量）

计算各链路的在线流量占链路百分比

记录各链路利用率到 txt 文件中

**数据处理**：原始在线流量数据记录的是某两个数据中心之间的在线流量传输，而在调度时用到的却是组网内各段链路上的在线流量，因此需要基于原始的在线流量数据来统计出各段链路上的在线流量。

1. recompute\_regions\_links（计算域间链路的在线流量）：

1.将所有经过该链路的路径的在线流量累加

2.将结果记录到 csv 文件

② recompute\_core\_links（计算域内链路的在线流量）：

1.遍历每个原始在线流量 csv 文件（不同路径），把该路径的流量分别累加到其经过的每一条链路上。

2.将计算好的各链路的在线流量记录到 csv 文件中

**预测在线流量**：

1. 从 csv 文件中读入各链路的在线流量
2. 计算各链路的真实剩余流量

1.真实剩余流量，即为某个周期中，某段链路的总流量减去在线流量之后所剩的那部分流量

2.预测剩余流量，即为某个周期中，预测算法计算得到的某段链路预计剩余的流量

1. 将原来 10min 为间隔的数据填充到 5s 为间隔
2. 计算各链路的预测剩余流量

1.贝叶斯拐点检测算法判断当前周期是否为拐点

2.k-Sigma 算法根据历史数据预测当前周期的剩余流量

3.Sliding k 算法权衡预测的稳定性和敏感程度

（1）若检测到当前周期为拐点，预测时更关注最近的数据，确保预测的铭感程度

（2）若当前周期不是拐点，预测时更多地关注历史数据，确保预测的稳定性

1. 录入初始真实剩余流量，初始预测剩余流量，在线流量到集中控制模块

有关贝叶斯拐点检测算法的原理可以看这个网址：<https://github.com/hildensia/bayesian_changepoint_detection>