有一个贪婪部署vnf的文件：main.py

这个文件使用了一个脚本，这个脚本是一张图，这个图中包含了42个节点66条链路。代码中生成了15条sfc，每个sfc有6个vnf，每个vnf不同类型，且不能被部署到同一个节点上，vnf按照资源最大和时延最小来部署，就是部署中vnf选择最大资源节点的来部署，虚拟链路选择延迟最小的最短路径来部署。部署成功后输出了两个csv文件，文件内容就是那42个节点的资源使用情况，和vnf的资源情况和部署情况。

（代码打印出来的东西是所生成的sfc链，包含的vnf和随机生成的资源，然后这个vnf部署到了哪个节点上，然后寻找最短路径最短时延和每个节点的资源使用情况）

我想基于这些数据，让您实现一个基于ddpg的vnf迁移算法，就是将过载节点（超过cpu总资源\*85%或者总内存资源\*85%的节点）上的vnf进行迁移，迁移到别的满足要求的目标节点上，

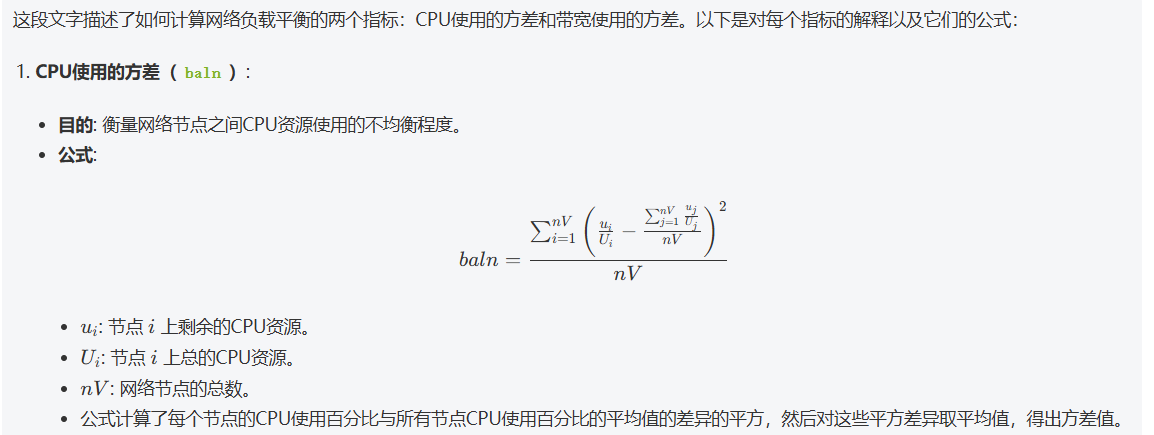
首先计算优先级：计算文件：score.py

根据计算的优先级对所要迁移的vnf实施迁移,如果不过载，就不继续迁移了

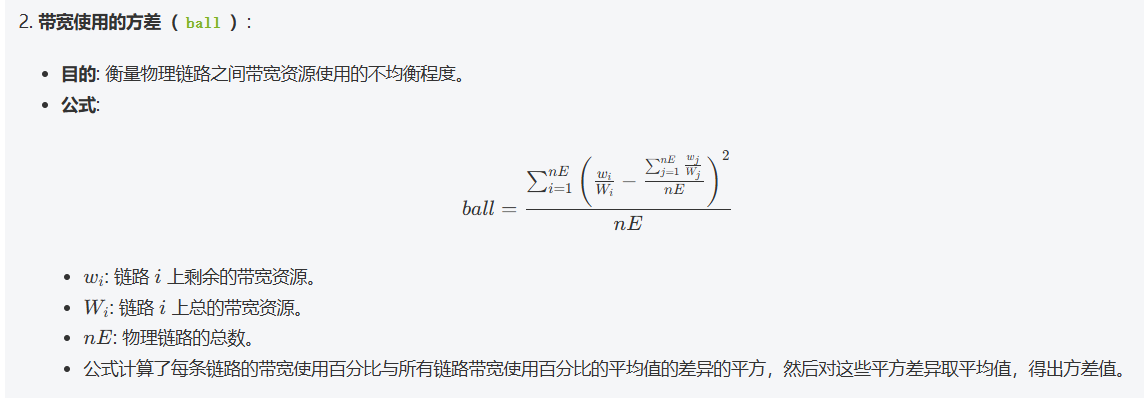
在迁移过程中，就是要满足这三点：

1. 负载均衡模型：这个目标节点要满足要迁移过来的vnf 的所有资源（cpu、内存），也就是加上迁移过来的vnf的资源还不过载。

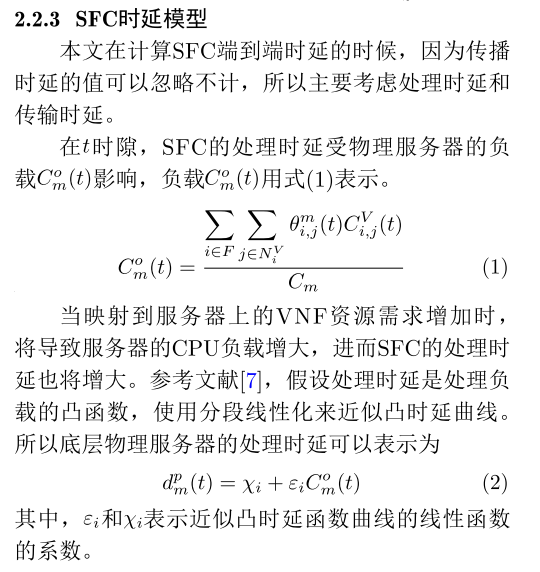
公式：



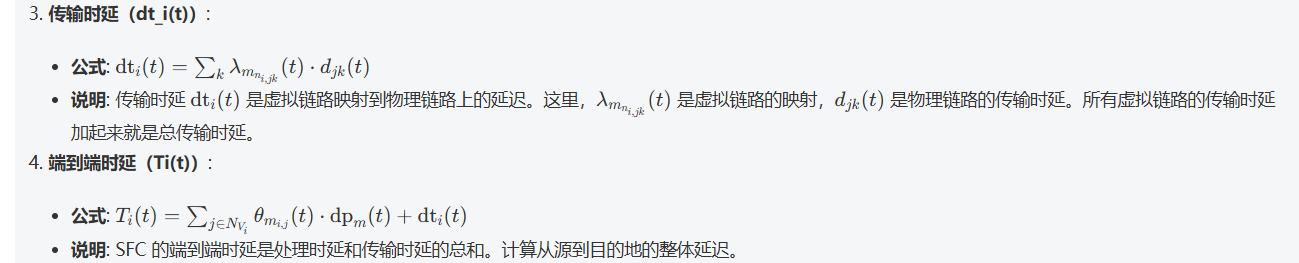
**内存的方差同cpu一样（同上公式）**



1. 时延模型：



SFC的传输时延与SFC上的虚拟链路映射到物理链路的位置有关，



T总时延=处理时延+传输时延

举例：

假设有一个SFC，它包括三个VNF，映射到两个物理服务器上。每个VNF的计算需求会影响服务器的负载，从而影响处理时延。如果VNF映射到服务器上的计算需求增加，服务器负载增加，从而处理时延也增加。

假设每个虚拟链路的传输延迟为10ms，而物理链路的处理时延为50ms。总的端到端时延计算为处理时延和所有虚拟链路的传输时延之和。假如总的处理时延为200ms，总的传输时延为30ms，则端到端时延为230ms。

奖励就是上述这些优化目标

Min a1\*负载均衡cpu+a2\*负载均衡内存+a3\*负载均衡带宽+a4\*总时延

解释：其实 sfc就是一条服务功能链，由vnf组成，并且依次相连。(VNF其实就是虚拟网络功能，就是比如说模拟路由器的功能，所以就是一个虚拟的路由器)。

Vnf1——vnf2——vnf3——vnf4——vnf5——vnf6

然后将这些vnf映射到节点上（服务器上）

比如由44个节点66条链路，就是把上述这个sfc也就是6个vnf放到这44个节点中的6个上，不能放到同一个节点上。但是这六个是依次相连的，所以这个虚拟的链路也要映射到物理链路上，我就用最短路径（这个最短路径的传播延迟最小）。这就是部署。

部署之后发现有过载节点，所以要将上面的vnf进行迁移出来，迁移到其他满足要求的节点上继续工作，并且确保这个sfc链还是连续的，且传播时延也短。