我目前边缘计算的结构如下：

一个Macro Base Station (MBS)区域内有多个Small Base Station (SBS)。

在SBS内的任务可以选择上传到SBS或者MBS，这是由SBS服务器和MBS的服务器状态所决定的。当SBS的服务器很拥挤时，而MBS的服务器空闲时，就可以将SBS区域内的任务上传到MBS。

这里的拥挤程度是用任务的***平均完成时间***和***任务的最晚完成时间（要求时延）***的比例来刻画的。由于任务到达时，服务器需要建立一个虚拟机来计算这个任务，那么这个虚拟机需要占据多少资源（CPU）是需要在任务到达后和计算前提前分配好的。如果任务到达的少，单个任务分配的资源越多，那么任务完成的时间就越短。那么就可以来刻画服务器的拥挤程度。

对于资源分配方面（CPU）有两个假设：第一个：服务器能够将到来的任务划分为多个不同计算类型的任务（人脸识别、VR、视频加速）。第二个：同一类的任务的计算所需要消耗的资源与任务本身的数据量有一定的关系。比如图像处理的人脸识别，假设要求运算时间一致，所需要的运算资源和图片的数据大小有一定的关系。

目前想做的服务器的任务的计算资源（CPU的计算能力的分配）分配是做一个ONLINE的分配，当一个任务到达时，服务器就会给任务分配一定的CPU，形成虚拟机来计算该任务。目前考虑的是，假设任务到达是依据一定的概率到达，依照前一段时间任务的到达实际情况，可以得出这个任务到达的快慢程度，根据这个到达模型来为任务分配资源。当任务到达频率很高是，服务器负载很大时，也可以根据这个概率来拒绝任务的上传。

目前看到斯坦福课程里有一个Online Decision-Making，在做决策是依据一定的概率做出决策，现在想法是能否应用到自己的这个模型上。