# 4

做调度其实是一个优化的过程，其目标函数从简单到复杂举例为：最简单的标准例如平均负载地理位置具有显而易见的调度方式，QoS，包括传输带宽，时延和抖动，丢包率等参数，某一个Session的视频QoS是确定的，无论是那个用户来看，而包括观看时长，主观评分等参数的QoE是指用户体验质量，不同用户对同一段视频的体验可能不同，甚至在不同的场合下都能对同一组QoS参数的Session有不同评价。比如某人在家里看视频和地铁中看视频相比对首帧卡顿的容忍率就会更低

右侧是E2E文章中网页加载时间对用户QoE的影响，可以看到明显不是线性的，而且因人而异

# 5

我们希望这个策略能够通过包括实时CDN信息与以及（用户信息、直播间信息）的二元组，来建立一个用户对session的偏好模型，这个模型能够综合考虑用户端的信息与CDN的信息对某个可能的分配进行打分，之后通过这个分数来决定CDN的分配

# 6

下面是一些前人的方案：

基于预测的调度策略大致流程为：首先要建立一个描述用户体验的优化的目标（或标准）与其计算方法，之后提出一个调度策略来试图在多个选择中选取能够最大化目标函数的

这篇是16年的文章，通过提取某个session的重要特征来与其他session聚类，从而对这个session的视频质量进行预测，通过预测的质量来匹配CDN

这篇文章，以及大部分CDN资源调度的文章没有考虑到用户对同一个session有不同评价的可能性，因为这里的视频质量完全由QoS决定，也就是说他自己有一个固定的QoS到QoE的映射，我们想要改进的正是这一点，我们希望考虑用户个人的喜好从而让资源分配有更高的用户体验上的收益

E2E是19年的一篇文章，它发现了不同用户处在非线性QoE曲线（这个曲线是通过数据得到的同一条的曲线，描述了平均上来说用户对页面加载速度的体验变化）的不同位置，花费同样资源情况下取得的用户体验的收益可能不同，因而当许多用户来发起请求的时候它通过两个步骤来做资源调度：

第一步将服务器端的slots分好，这样第二部无论如何分配用户都不会影响到既定的服务器性能，第二部做最大权重和的二分图匹配，将QoE函数的和作为目标函数得到最大值从而为用户分配slots。这里的QoE是由用户的状态与服务器的状态共同决定的，考虑到了用户的主观感受。但这里是以数据库来举例，我们在为视频来匹配CDN的时候需要考虑相比仅一个延迟来说更多更高维的参数。

# 8

QoE的metric我们按照广为接受的标准定为观看时长，在两个不同的QoS参数下，均符合预期，在加载时间很短的时候人们感觉不到太大差异，加载时间较长则变化放缓。故如果分配资源的系统相对更多地考虑处在快速变化区的用户，总QoE提升空间较大。值得一提的是，我们的ffm模型预期能够对多个QoS参数综合考虑给出QoE预测值，绝不仅仅是对一维特征的函数拟合再加权。

# 9

不同用户之间对于同样的QoS标准（横轴），QoE（纵轴）的变化方式不尽相同

对于不同直播间来说同理

这是符合直觉的，比如不同种类的直播间，播户外的对于画质要求就不算高，但游戏直播间的观众则不能忍受低画质

# 10

在session层面上根据偏好模型中训练的embedding vector做聚类，如果类数不多可以做到预测下一个时段每一类将会有多少用户（简单实现可以直接将前几天的做平均，可能周末或平时有差别，但是有规律可循，且较为稳定）

决定每个CDN的预期负载。最简单的方法可以grid search枚举，每一个分配方式当中CDN的负载固定，于是QoS参数是确定的，user class本来就是确定的，因此就可以计算每条边的边权来进行最大权重最大流算法，在所有枚举的分配方式中选取权重最大的网络流即代表了接下来一段时间内的调度比例

# 11

用户偏好模型可以用推荐系统