

2022 华为软件精英挑战赛

复赛任务书(练习赛)

文档版本

01

发布日期

2022-03-31



修订记录

版本	发布日期	修改说明
01	2022-03-31	第一次正式发布。

目 录

修订记录.....	ii
1 背景信息.....	1
2 题目.....	2
2.1 题目介绍.....	2
2.2 题目数学描述.....	4
3 输入说明.....	6
4 输出说明.....	8
5 评分规则.....	9

1 背景信息

提升用户体验的同时降低运营成本是云服务竞争力的关键。

在视频直播场景中，网络成本是影响服务成本的关键因素之一，不同的流量调度方案会产生不同的网络使用成本。本赛题以华为云视频直播服务流量调度问题为基础，并进行一定的抽象、调整和简化。参赛选手需要设计高效的调度算法，在满足客户要求的前提下，通过对流量的合理调度，最小化网络使用成本。

期待您的精彩解决方案。

2 题目

2.1 题目介绍

术语

表2-1 术语

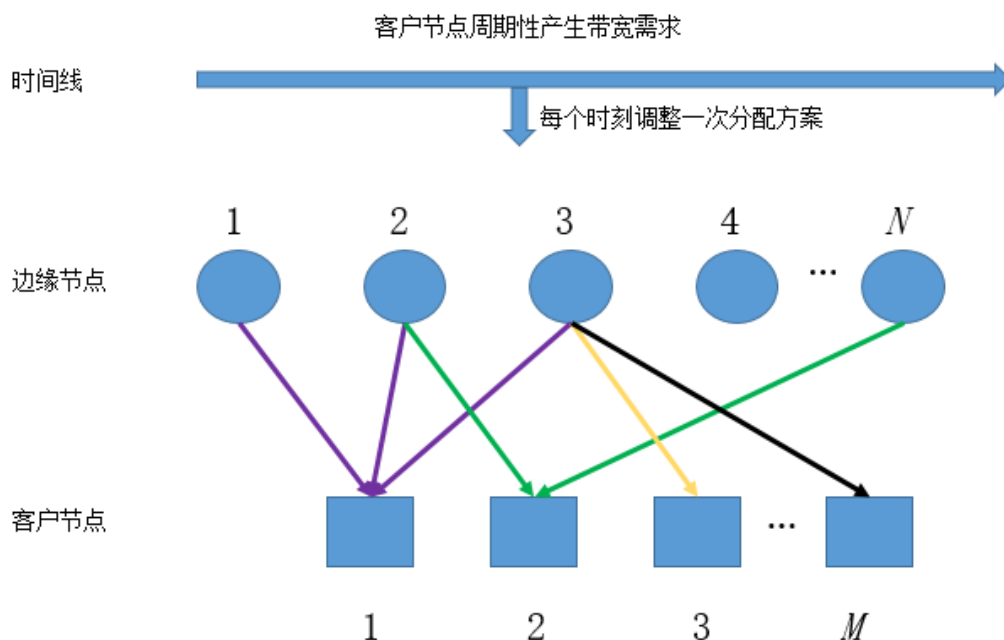
名词	解释
时刻	流量调度沿时间线进行，将时间线划分成若干个独立的调度单元，用单元的起始时间来指代这个单元，称为时刻。
带宽	在一个时刻，某个节点的流量大小称为带宽。
流	一个视频直播的数据流。
客户节点	一种抽象节点，在每个时刻，每个客户节点会产生新的带宽需求。
边缘节点	一种抽象节点，和客户节点直接交互的网络节点。客户节点通过边缘节点获取数据流量。在每个时刻，每个边缘节点可以为若干客户节点提供带宽，但它能提供的带宽之和有上限。在每个时刻，一个客户节点的带宽需求可以分配到一个或多个边缘节点。
QoS	客户节点和边缘节点之间的网络质量。不同客户节点和边缘节点之间的网络会因距离、网络连接状况有差别。本题中 QoS 简化成一个数值，表示时延大小（单位：ms）。
边缘节点带宽序列	对于某个边缘节点，统计它在每个时刻分配的带宽之和，这些值在整个时间线上形成一个带宽序列。
95 百分位带宽	<p>对一个边缘节点带宽序列进行升序排序，以其 95%位置（向上取整）的带宽值作为该节点的 95 百分位带宽。</p> <p>示例：假设一个节点的带宽序列长度为 8928。首先对这 8928 个带宽值升序排列。其 95%位置是 8481.6，向上取整为 8482。则以排序序列中第 8482 个点的带宽值，作为这个节点的 95 百分位带宽。</p>

名词	解释
边缘节点成本	基于每个边缘节点的 95 百分位带宽值，按照题目数学描述部分的成本计算公式，得到边缘节点成本。
带宽总成本	基于边缘节点成本，按照题目数学描述部分的成本计算公式，得到带宽总成本。

题目描述

某个时刻边缘节点为客户节点分配的带宽资源示意图如图 2-1 示。

图2-1 某时刻带宽分配示意图



1. 共有 M 个客户节点和 N 个边缘节点。
2. 在每个时刻，要决策如何把每个客户节点的带宽需求分配到边缘节点。
3. 在每个时刻，会有若干种流。
4. 在每个时刻，每个客户节点对于每种流会有带宽需求（0 表示该时刻该客户节点对这种流无带宽需求）。为了实现流的流向端到端可追溯，在每个时刻，一个客户节点对一种流的带宽需求需要不可拆分地分配到一个边缘节点。
5. 为了确保调度质量，每个客户节点的需求只能分配到满足 QoS 约束的边缘节点上。即：当客户节点和边缘节点之间的 QoS 小于“QoS 上限”时，才会进行流量分配。
6. 在每个时刻，每个边缘节点接收的带宽需求总和不能超过其带宽上限。

7. 合理分配所有时刻的客户节点带宽需求, 使得最终的带宽总成本尽量小。

2.2 题目数学描述

题目的数学模型抽象如下:

定义

1. 集合 T ($1 < |T| \leq 8928$) 为所有时刻的集合。
2. 共有 M ($1 \leq M \leq 35$) 个客户节点。
3. 共有 N ($1 \leq N \leq 135$) 个边缘节点, 第 j 个边缘节点在任何时刻的带宽上限都是 C_j ($1 \leq j \leq N$)。
4. 第 i 个客户节点与第 j 个边缘节点之间的 QoS 为 Y_{ij} ($1 \leq i \leq M, 1 \leq j \leq N$)。
5. QoS 约束上限为 Q (从配置文件读取)。
6. t 时刻流的种类数为 p^t ($1 \leq p^t \leq 100, t \in T$)。
7. t 时刻第 i 个客户节点的第 k 种流的带宽需求值为 d_{ik}^t ($1 \leq i \leq M, 1 \leq k \leq p^t, t \in T$); 如果 t 时刻第 i 个客户节点对第 k 种流无带宽需求, 则 $d_{ik}^t = 0$ 。
8. 用 X 表示一组流量分配方案, 即 t 时刻第 i 个客户节点的第 k 种流是否分配给第 j 个边缘节点表示为 X_{ijk}^t ($1 \leq i \leq M, 1 \leq j \leq N, 1 \leq k \leq p^t, t \in T$)。 X_{ijk}^t 取值范围为 $\{0,1\}$, 0 表示不分配, 1 表示分配。
9. t 时刻第 j 个边缘节点接收的带宽需求为 $w_j^t = \sum_{i=1}^M \sum_{k=1}^{p^t} X_{ijk}^t d_{ik}^t$ ($1 \leq j \leq N, t \in T$)。对所有 $t \in T$, 计算边缘节点 j 的 95 百分位带宽值为 W_j 。
10. 边缘节点成本调整参数为 V (从配置文件读取)。
11. 第 j 个边缘节点成本为 s_j ($1 \leq j \leq N$):

$$s_j = \begin{cases} 0, \sum_{t \in T} w_j^t = 0 \\ V, \sum_{t \in T} w_j^t > 0 \wedge 0 \leq W_j \leq V \\ \frac{1}{C_j} (W_j - V)^2 + W_j, W_j > V \end{cases}$$

12. 带宽总成本为 $Cost$ 。

约束

1. $\forall t \in T, 1 \leq i \leq M, 1 \leq j \leq N, 1 \leq k \leq p^t, d_{ik}^t = 0 \quad X_{ijk}^t = 0$
带宽需求大小为 0 的流不分配给任何边缘节点。
2. $\forall t \in T, 1 \leq i \leq M, 1 \leq j \leq N, 1 \leq k \leq p^t, Y_{ij} \geq Q \quad X_{ijk}^t = 0$
客户节点带宽需求只能分配到满足 QoS 约束的边缘节点。
3. $\forall t \in T, 1 \leq i \leq M, 1 \leq k \leq p^t, d_{ik}^t > 0 \quad \sum_{j=1}^N X_{ijk}^t = 1$
每个时刻, 每个客户节点的每种流的带宽需求只能分配给一个边缘节点。
4. $\forall t \in T, 1 \leq j \leq N \quad w_j^t \leq C_j$ 。
边缘节点接收的带宽需求不能超过其带宽上限。

优化目标

$$Cost = \min_X \left(\left\lfloor \sum_{j=1}^N s_j + 0.5 \right\rfloor \right)$$

找到一组满足约束的流量分配方案 X ，使其在时间集合 T 内的带宽总成本尽可能小。因为边缘节点成本计算公式会产生小数，约定计算过程中 s_j 使用 64 位浮点数计算，对 s_j 求和后四舍五入得到整数值，作为最终的带宽总成本。

须知

正式比赛阶段，会在本复赛任务书的赛题基础上增加小的变更点。选手需要在复赛正式赛时调整代码，根据新增条件完成求解。变更点详见正式比赛时发布的《2022 华为软件精英挑战赛复赛变更点》。

3 输入说明

线上评测时，输入数据从/data 目录读取，涉及多个文件，文件的换行符为\r\n。

1. **demand.csv**: 以逗号分割的 CSV 文件，包含所有客户节点在不同时刻的带宽需求信息。数据含义如下：
 - **第 1 列**: 即 **mtime** 列，表示不同时刻。
 - **第 2 列**: 即 **stream_id** 列，表示流 ID，标识一个流。长度不超过 3 的字符串，由大小写字母或数字组成。在每个时刻，流 ID 是唯一的。
 - **第 3 列~第 $M + 2$ 列**:
 - 第 1 行: 表示客户节点 ID，唯一标识一个客户节点。长度不超过 2 的字符串，由大小写字母或数字组成。
 - 第 2 行~最后一行: 每一个数值表示该客户节点在一个时刻对一个流的带宽需求大小。非负整数，单位 MB。
 - **客户节点数**: $M \leq 35$ 。
 - **带宽值**: 不超过 80,000MB。
 - **每个时刻流的种类数**: ≤ 100 。
 - **demand.csv**: 文件大小不超过 200MB。
 - 一个 $|T| = 2, M = 3$ ，两个时刻的流种类数都为 2 的例子如下:

```
mtime,stream_id,CB,CA,CE
2021-10-19T00:00,XH,6056,7800,0
2021-10-19T00:00,XP,5120,900,0
2021-10-19T00:05,XH,5890,400,0
2021-10-19T00:05,XU,600,890,234
```

2. **site_bandwidth.csv**: 以逗号分割的 CSV 文件，包含边缘节点列表以及每个边缘节点的带宽上限。数据含义如下：
 - **第 1 列**: 即 **site_name** 列，表示边缘节点 ID，唯一标识一个边缘节点。长度不超过 2 的字符串，由大小写字母或数字组成。
 - **第 2 列**: 即 **bandwidth** 列，表示对应边缘节点的带宽上限。非负整数，单位为 MB。
 - **边缘节点数**: $N \leq 135$
 - **带宽值**: 不超过 1,000,000MB。

- 一个 $N = 3$ 的例子如下:

```
site name,bandwidth
S1,81920
SB,87040
SD,74586
```

3. qos.csv: 以逗号分割的 CSV 文件, 包含客户节点与边缘节点之间的网络时延。数据含义如下:

- **第 1 列:** 即 site_name 列, 表示边缘节点 ID。
- **第 2 列~第 $M + 1$ 列:**
 - **第 1 行:** 表示客户节点的 ID。
 - **第 2 行~最后一行:** 边缘节点到客户节点的时延, 非负整数, 单位为 ms。
- **网络时延取值范围:** $0 < \text{QoS} \leq 1000\text{ms}$ 。
- 一个 $M = 3, N = 3$ 的例子如下:

```
site name,CB,CA,CE
S1,200,186,123
SB,190,48,23
SD,340,45,56
```

4. config.ini: 参数配置文件。格式为:

- **[config]:** 参数配置开头
- **qos_constraint=value1:** QoS 约束上限 Q 的配置信息, value1 为非负整数, 单位为 ms。
- **base_cost=value2:** 边缘节点成本调整参数 V 的配置信息, value2 为非负整数。

假设配置 QoS 约束上限值为 400ms, 边缘节点成本调整参数为 400, 那么配置文件示例如下:

```
[config]
qos_constraint=400
base_cost=400
```

4 输出说明

输出说明

线上评测时，带宽的调度分配方案输出路径为：/output/solution.txt。

输出要求：

1. 按照输入文件的时刻顺序，输出分配方案。
2. 每个时刻，所有客户节点的带宽需求分配方案用 M 行表示。每行表示一个客户节点的带宽需求分配方案，每行的格式为：

```
customer_ID1:<site_id_1,stream_id_1,stream_id_2,...>,<site_id_2,stream_id_3,stream_id_4,...>...
```

其中，

- **customer_ID1**: 表示客户节点 ID。
- **stream_id_1,stream_id_2,...**: 代表多个流的 ID。
- **<site_id_1,stream_id_1,stream_id_2,...>**: 表示把多个流的带宽需求分配给 ID 为 site_id_1 的边缘节点。

3. 如果在某个时刻，一个客户节点的带宽需求为 0（表示没有带宽需求），则只输出客户 ID。格式为：

```
customer_ID1:
```

4. 每个时刻，客户节点的顺序不做要求；每一行内，边缘节点顺序不做要求。
5. 最终输出文件包含 $|T| \times M$ 行，换行符为\r\n 或\n，不允许有空行。

输出示例

以前面输入说明示例数据为基础，在 QoS 约束上限为 400ms 的情况下，一组流量分配方案为：

```
CB:<S1,XH>,<SB,XP>  
CA:<S1,XH>,<SD,XP>  
CE:  
CB:<S1,XH,XU>  
CA:<SB,XH>,<SD,XU>  
CE:<SD,XU>
```

5 评分规则

1. 判题程序会从选手程序输出的 `solution.txt` 文件读取分配方案，计算带宽总成本并记录程序运行时间（单位为 ms）。
2. 带宽总成本低的方案胜出。
3. 如果不同选手的输出方案的总成本相同，则运行时间少者胜出；如果运行时间也相同，则先提交代码者胜出。
4. 若采用多组数据评分，则多组数据的带宽成本与运行时间分别求和后排名。
5. 对于每组数据，选手的程序所有计算步骤（包含读取输入、计算、输出方案）所用时间总和不超过 300 秒。若程序运行超时、运行出错或输出不合法的解（包括调度分配方案不满足题目约束或解格式不正确），则判定无成绩。
6. 对于多组数据，选手的程序在任意一组数据上无成绩，则判定整体无成绩。

注意

因为进程调用存在一定的时间开销，用时统计在判题程序侧和选手程序侧可能存在细微差异。建议选手控制算法用时的时候要留有一定的冗余。