Optimization Issues in Multimedia Systems

Batching Optimization

- Instead of allocating a single I/O resource per client, we can reduce the I/O demand (or the system can support more clients) by sharing one I/O resource.
- Assume we have N videos with f_i (or $f_i = \lambda p_i$) being the request frequency for movie i^{th}. We let S be the server capacity (i.e., number of streams that can be supported). Let L_i denotes the length of the video i. The average video length is

$$L = \frac{\sum_{i=1}^{N} f_i L_i}{\sum_{i=1}^{N} f_i}$$

- The average number of streams that can be scheduled by the server *per unit time* at full capacity is S/L.

- We the following *design variables*, $t_1, t_2, \cdots, t_N$ be the *average time intervals* that video $1, 2, \cdots, N$ are batched.
- The average startup latency for video i is $t_i/2$.
- The objective is to *minimize* the expected startup latency of any client, or

$$\begin{aligned} \min f(t_1, \cdots, t_N) &= \sum_{i=1}^{N} f_i \frac{t_i}{2} \\ \text{s.t.} \qquad \sum_{i=1}^{N} 1/t_i &= S/L \\ t_i &\geq 0 \qquad \forall i \end{aligned} \tag{1}$$

- Due to the nature of the problem, the *Kuhn-Tucker* conditions are the necessary and sufficient conditions for the existence of a *global optimum solution.* The K-T conditions are:

$$
\begin{aligned}
f_i - u/t_i^2 &= 0 \\
\text{and} \qquad \sum_{i=1}^{N} 1/t_i &= S/L
\end{aligned}
\tag{2}
$$

where u is the *Lagrange multiplier.*

- The first equation above implies the following *invariant*:

$$
t_1\sqrt{f_1} = t_2\sqrt{f_2} = \quad \cdots \quad = t_N\sqrt{f_N} \tag{3}
$$

- The question is, what should these t_i be?

- Let q_i be the average queue length of the i^{th} video *at the time of batching.* Using the Little's result, we can estimate t_i by

$$
q_i \quad = \quad f_i t_i \tag{4}
$$

- Therefore, we have the following invariant:

$$\frac{q_1}{\sqrt{f_i}} = \frac{q_2}{\sqrt{f_2}} = \quad \cdots \quad \frac{q_N}{\sqrt{f_N}} \tag{5}$$

- Define the *factor queue length* of video i be $\frac{q_i}{\sqrt{\lambda p_i}}$.

- The *intuitive result* is that to minimize the average startup latency, *whenever a stream becomes available, schedule the video with the LARGEST factored queue length.*

- This implies we need to robustly measure f_i. Let $f_i \approx q_i/\delta t$ where δt is the interval since the last time that video i was scheduled. We have the following result: *whenever a stream become available, schedule the video with the LARGEST value of $(q_i \delta t)$.*