WFQ IS-based Simulation

Xinyun Chen, Jin Guang, Guiyu Hong
School of Data Science
Chinese University of Hong Kong, Shenzhen

Contents

- The flow of the WFQ scheduling algorithm
- Algorithm design for the WFQ IS-based simulation
  - Regenerative method
  - Importance sampling
  - Cross-entropy method
  - Quantile Estimation

WFQ Scheduling Algorithm: Introduction

- WFQ is proposed by Demers, Keshav, and Shenker 1989* as a natural extension of bit-by-bit round robin.
- Given that a scheduler handling K flows is configured with one weight w_k for each flow, the flow k will achieve an average data rate of
$$\frac{w_k}{w_1 + w_2 + \cdots + w_K}\mu,$$
where μ is the link rate.

** Demers, Alan, Srinivasan Keshav, and Scott Shenker (1989). "Analysis and simulation of a fair queueing algorithm". In: ACM SIGCOMM Computer Communication Review 19.4, pp. 1-12.*

WFQ: Bit-by-Bit Round Robin

- Assumptions:
  - The scheduler transmits the data bit-by-bit
  - For each round, the flow k can transmits w_k bits (suppose w_k is an integer)
- Notations:
  - $N(t)$: the number of active flow at time t.
  - $V(t)$: the number of rounds made in the round-robin service discipline up to time t. Note that it is set as a continuous function, with the fractional part indicating partially completed rounds.
- Key: Virtual Time $V(t)$

  If $N(t)$ is unchanged during time $[t_1, t_2)$, the increment of the number of rounds is

$$V(t_2) - V(t_1) = \frac{\mu}{\sum_{j \in N(t_1)} w_j}(t_2 - t_1)$$

WFQ Scheduling Algorithm

- Define:
  - $T_{k,n}$: arrival time of packet n of flow k
  - $S_{k,n}$: packet size (in bits) of packet n of flow k
  - $\widetilde{D}_{k,n}$: virtual departure time (in round number) of packet n of flow k
- Relation:

$$\widetilde{D}_{k,n+1} = \max(\widetilde{D}_{k,n}, V(T_{k,n+1})) + \frac{S_{k,n+1}}{w_k}$$

- From Bit-by-Bit Round Robin to Weighted Fair Queue:
  - Select the packet with smallest virtual finishing time.

WFQ Simulation: Problem Setting

- Assumptions:
  - Arrival process: Poisson process with the rate λ_k.
  - Packet size distribution: exponential distribution with mean $1/\mu_k$ bit.
- Target: the p-tile of steady-state response time of class-k jobs

$$Q_k(p) = \inf\{\gamma : P\left(R_{k,\infty} > \gamma\right) < 1 - p\}$$

WFQ Simulation: Regenerative Method

- Similarly as the priority queue, we define that the system regenerates whenever a job finds the server idle upon departure.
- Recall that the regenerative method is

$$P\left(R_{k,\infty} > \gamma\right) = \frac{E\left[\sum_{n=1}^{\alpha_k} I\left(R_{k,n} > \gamma\right)\right]}{E\left[\alpha_k\right]}$$

- The denominator can be estimated accurately by naïve simulation, but the nominator can be improved by importance sampling.

WFQ Simulation: Importance Sampling

- Switching change of measure:
  - Original densities: f_k^A, f_k^S. IS densities: $\tilde{f}_k^A, \tilde{f}_k^S$.
  - Stage 1: Starting from time 0, we generate the inter-arrival times and service times according to some IS densities $\tilde{f}_k^A, \tilde{f}_k^S$ and simulate the system dynamics accordingly.
  - Stage 2: the simulator will switch back to the original distribution f_k^A, f_k^S once the event of interest $\{R_{k,n} > \gamma_{\max}\}$ is observed.
  - Define $R'_{k,n}$ as the sum of service times of all jobs who have less virtual departure time than itself $\widetilde{D}_{k,n}$ and the remaining service time of the job in service. And define a stopping time

$$\tau_k = \inf\{T_{k,n}: R'_{k,n} > \gamma_{\max}\}$$

WFQ Simulation: Importance Sampling

- Notations:
  - $G_l(\cdot)$: the tail probability of inter-arrival times in flow l.
  - $N_l(t)$: the number of job arrivals of class l by time t.
- The likelihood ratio upon time t can be computed as

$$L(t) = \frac{\prod_{l=1}^{K}\left(\prod_{n=1}^{N_l(t\wedge\tau_k)} \tilde{f}_l^A(A_{l,n})\tilde{f}_l^S(S_{l,n}) \cdot G_l(H_l)\right)}{\prod_{l=1}^{K}\left(\prod_{n=1}^{N_l(t\wedge\tau_k)} \tilde{f}_l^A(A_{l,n})\tilde{f}_l^S(S_{l,n}) \cdot \tilde{G}_l(H_l)\right)}$$

where H_l is the age since last arrival of class l at time $t \wedge \tau_k$.
- Importance Sampling for nominator:
  - $E\left[\sum_{n=1}^{\alpha_k} I(R_{k,n} > \gamma)\right] = \tilde{E}\left[\sum_{n=1}^{\alpha_k} I(R_{k,n} > \gamma)L(E_{k,n})\right]$ where $E_{k,n}$ is starting service time.

WFQ Simulation: Cross-Entropy Method

- The optimization problem is

$$\max_{f'} E\left[\sum_{n=1}^{\alpha_k} I\left(R_{k,n} > \gamma\right) \cdot \log L'(\alpha)\right]$$

  - The adaptive cross-entropy method can also be used.

WFQ Simulation: Quantile Estimator

- Sample m_1 cycles under the IS distribution and get response times $\left\{\tilde{R}_{k,1}, \dots, \tilde{R}_{k,\tilde{\beta}_{m_1}}\right\}$. Also, sample m_2 cycles under the original distribution.
- The quantile estimator is

$$\hat{Q}_{m_1,m_2}(p) = \inf\{\gamma : \hat{P}_{m_1,m_2}(R_{k,\infty} > \gamma) < 1 - p\}$$
$$= \inf\left\{\gamma : \sum_{i=1}^{\tilde{\beta}_{m_1}} L_{k,i} I\left(\tilde{R}_{k,\infty} > \gamma\right) < \frac{(1-p)m_1}{m_2} \sum_{m=1}^{m_2} \alpha_k^i\right\}$$

- To get $\hat{Q}(p)$, we can sort $(\tilde{R}_1, \tilde{R}_2, \dots, \tilde{R}_{\tilde{\beta}_{m_1}})$ in ascending order, and

$$\hat{Q}_{m_1,m_2}(p) = \tilde{R}_{k,(n_p)} \quad and \quad n_p = \min\left\{n : \sum_{i=n}^{\tilde{\beta}_{m_1}} L_{k,(i)} < \frac{(1-p)m_1}{m_2} \sum_{m=1}^{m_2} \alpha_k^i\right\}$$