Computer Time-Sharing and Multi-access Systems

Professor Yeali S. Sunny
Department of Information Management
National Taiwan University
Outline

- Computer system structure as a feedback queueing model
- The attained service (response time) (4.2)
- The batch processing algorithm (4.3)
- The round-robin scheduling algorithm (time-sharing) (4.4)
- Finite population models (4.11)
Computer System Structure

A system of resources

Figure 4.1 Computer system structure, (—— denotes data flow, ---- denotes control).
Introduction

- A number of users requesting access to resources (e.g., computing, database, etc.)

- **Scalability**
  - maximum throughput capacity
  - Bottleneck resource
  - User’s perspective performance

- The problem of **resource sharing**
  - Fairness
  - Different users have different needs
  - Conflict among users for simultaneous access to the system.
Introduction (cont'd)

- To resolve the conflict -> scheduling algorithm is required.
- Determine which user to serve next and for how long
- Single resource (e.g., a CPU)
- Multiple resources
Time-sharing system as a feedback queueing system

- “Quantum”, depart or re-enter
- Can be seen as a preemptive resume priority queueing rule.
- It permit the use of the computer by many users simultaneously -> resource sharing.
- The system gives preferential treatment to short jobs at the expense of the longer ones.
How to differentiate out short jobs

- In reality, the system usually does not know the exact service time each arriving customer places on the system.

- Typically, the distribution of the service time is given.

- Assume negligible processing time (e.g., job swap in and out)

- How can we service the jobs giving priority in relation to their demands?
Time sharing system

Notations

- $A(t)$: interarrival time distribution of customers (jobs)
- $B(x)$: service time
- $q_{pn}$: the quantum to a customer of priority $p$ on his $n^{th}$ entry to the system (4.1)
Time sharing system (cont’d)

Want to find the system time distribution (i.e. response time) for a job requiring service time $x$

\[
S(y \mid x) = \Delta P[\text{response time} \leq y \mid \tilde{x} = x] \quad (4.2)
\]

\[
T(x) = \Delta \text{ average response time for a customer requiring } x \text{ sec of processing} \quad (4.3)
\]

$W(x) = T(x) - x$ -> average “wasted” time (the time to sort out it has the shortest service time w.r.t. ideal shortest-job-first system).
Derivation of the attained service time distribution (1/5)

- Given service time distribution
- Negligible swap time
- Assume all quanta shrink to zero
  - When all quanta shrink to zero, it results in “processor-sharing” system
  - The “processor-sharing” system gives good mathematical approximation to the finite quantum results.
- Consider M/G/1 with $\lambda$, $x$ and $\rho = \frac{\lambda}{x} < 1$
- Results are good as well for G/G/1.
Derivation of the attained service time distribution (2/5)

- At any time instant, number of jobs and individual amount of time received so far (i.e. attained service)
- \( n(x) \) – average density of the number of customers so far received \( x \) sec of service (4.5)
- (4.6) - average number of customers so far received between \( x_1 \) and \( x_2 \) sec of service
- \( N(x) \) - average number of customers whose attained service is exactly \( x \) sec (4.7)
Derivation of the attained service time distribution (3/5)

- $q_n$: the quantum to a customer at his $n^{th}$ service entry
- $Q_n$: total service amount received for a customer have visited the server exactly $n$ times.

For customers with service requirement $x$, $Q_{n-1} < x \leq Q_n$, we have $W(x) = W(Q_n)$

- $W(Q_n)$: average time in the system prior to the $n^{th}$ visit with $x > Q_{n-1}$
Derivation of the attained service time distribution (4/5)

- $W(Q_n) - W(Q_{n-1})$: the average time in the system after the $(n-1)^{st}$ but prior to the $n^{th}$ visit.
- $[1 - B(Q_n)]$ – average arrival rate of jobs back to the system after having $n$ visits.
- By Little’s result, we have the expected number of customers in the system with attained service $Q_n$

$$N(Q_n) = [1 - B(Q_n)][W(Q_{n+1}) - W(Q_n)] \quad (4.9)$$
Derivation of the attained service time distribution (5/5)

- Let quanta approach zero, i.e., \( q \rightarrow 0 \)

\[
n(x) = \lambda [1 - B(x)] \frac{dT(x)}{dx} \quad (4.11)
\]

Inverse of the Average rate of service to a customer with an attained service of \( x \) sec.
The batch processing algorithm
For a Time Sharing System

Goal: find the system time distribution (i.e. response time) for a job requiring service time $x$

$$S(y \mid x) = P[\text{response time } \leq y \mid \tilde{x} = x] \quad (4.2)$$

How much time a job can receive on each server visit?
Resource Sharing

- **Time-sharing**
  - Customers are given the **full** capacity of the processor on a **part-time basis**.

- **Processor-sharing**
  - Customers are given a **fractional-capacity** processor on a **full-time basis**.
The Length of the “Quantum”

- **Batch processing system**
  - A simple way of resource sharing
  - A FCFS queueing system with *infinity* service quantum

- **Processor sharing system**
  - Quantum approximates *Infinitesimal*

- **Consider batch processing system as a FCFS system with an infinitesimal \((0^+)\) quantum**
  - After the termination of the quantum, the customer is immediately fed back to the head of the line and taken back to service)
A batch processing system modeled as a FCFS system with an infinitesimal quantum

Figure 4.3  The FCFS system with an infinitesimal quantum.
Recall ... the distribution of the number of customers in the system

- For M/G/1 system with service order independent of service time

\[ Q(z) = B^* (\lambda - \lambda z) \frac{(1 - p)(1 - z)}{B^* (\lambda - \lambda z) - z} \]  

(3.15)

- With FCFS, the Laplace transform of waiting time distribution

\[ W^*(s) = \frac{s(1 - \rho)}{s - \lambda + \lambda B^*(s)} \]
A FCFS System with an Infinitesimal Quantum

Find the response time distribution

Let

\[ S(y \mid x) = P[\text{response time} \leq y \mid \tilde{x} = x] \]  (4.2)

\& Laplace transform

\[ S^*(y \mid x) = \int_0^\infty e^{-sy} dS(y \mid x) \]

We know \( S(y \mid x) = W(y \mid x) + x \) and for M/G/1 system with FCFS,

\[ W^*(s) = \frac{s(1 - \rho)}{s - \lambda + \lambda B^*(s)} \]

\[ S^*(s \mid x) = \frac{s(1 - \rho)e^{-sx}}{s - \lambda + \lambda B^*(s)} \]  (4.12)
Batch Processing: Discussion

\[ S^*(s \mid x) = \frac{s(1 - \rho)e^{-sx}}{s - \lambda + \lambda B^*(s)} \]  \hspace{1cm} (4.12)

\[ T(x) = \frac{W_0}{1 - \rho} + x \]  \hspace{1cm} (4.13) \hspace{1cm} \text{where} \hspace{1cm} W_0 = \lambda \overline{x^2} / 2

- Offers no discrimination among jobs on the basis of their required service time.
- The average waiting time is independent of service time.
- The least discriminatory system possible!
- Common in many service provisioning in human societies.
The round-robin scheduling algorithm
Round-Robin System

- $q_{pn}$: the quantum to a customer of priority $p$ on his $n^{th}$ entry to the system
- Let $q_{pn} = q \rightarrow 0$
- Single queue
- Queueing discipline
  - FCFS and
  - Quantum expires and needs more service, return to the TAIL of queue
Derive system time for M/G/1 (1/3)

- Consider the rate of completing service (failure rate) given an attained service (age) of \( x \) sec.
  \[ \mu(x) = \frac{b(x)}{1 - B(x)} \quad (4.14) \]

- \( \Box (x) = \text{prob}[x \leq y < x + \Delta x | y > x] \)

- \( n(x) \) - the average density of customers in the system with an attained service of \( x \) sec

- \( n(x + \Delta x) = n(x)[1 - \mu(x)\Delta x] + o(\Delta x) \)

- When \( \Delta x \to 0 \), we have
  \[ \frac{dn(x)}{dx} = -\mu(x)n(x) \]

- The solution
  \[ n(x) = n(0) \exp[-\int_0^x \mu(y)dy] \]

  \[ -\int_0^x \mu(y)dy = \log_e[1 - B(x)] \]
Derive system time for $M/G/1$ \((1/3)\)

- Consider the rate of completing service (failure rate) given an attained service (age) of \(x\) sec.
  \[ \mu(x) = \frac{b(x)}{1 - B(x)} \quad (4.14) \]
- \(\Box (x) = \text{prob}[x \leq y < x + \Delta x | y > x]\)
- \(n(x)\) - the average density of customers in the system with an attained service of \(x\) sec
- \(n(x + \Delta x) = \text{Prob}[y > x + \Delta x | y > x]\)
- \(n(x + \Delta x) = n(x)\left[1 - \mu(x)\Delta x\right] + o(\Delta x)\)
- When \(\Box x \to 0\), we have
  \[ \frac{dn(x)}{dx} = -\mu(x)n(x) \]
- The solution \(a\)
  \[ n(x) = n(0)\exp[-\int_{0}^{x} \mu(y)dy] \]
- We know
  \[ -\int_{0}^{x} \mu(y)dy = \log_{e}[1 - B(x)] \]
- Take into \(a\) we obtain
  \[ n(x) = n(0)[1 - B(x)] \quad (4.15) \]
Derive system time for $M/G/1$ (2/3)

- (4.11) = (4.15)

\[ n(x) = \lambda[1 - B(x)] \frac{dT(x)}{dx} \quad (4.11) \]

We have

\[ n(x) = n(0)[1 - B(x)] \quad (4.15) \]

- We know $T(0) = 0$

- Then we have

\[ T(x) = \frac{n(0)}{\lambda} x \quad x \geq 0 \quad (4.16) \]

- We have

\[ \frac{dT(x)}{dx} = \frac{n(0)}{\lambda} \]
Derive system time for \( M/G/1 \) (3/3)

- Find \( n(0) \) - the average density of customers in the system with zero attained service.
- Consider a “test” job with \textit{infinite} service time.
- Any job will depart the system before the test job.
- Equivalent to the lowest priority customer in a preemptive HOL priority.
- The average waiting time for job of priority class \( p \)

\[
\begin{align*}
T_p &= \frac{x_p (1 - \sigma_p) + \sum_{i=p}^{P} \lambda_i x_i^2}{(1 - \sigma_p)(1 - \sigma_{p+1})} \\
&= \frac{\sum_{i=p}^{P} \rho_i}{1 - \sigma_p} \\
\end{align*}
\]

\[
\sigma_p = \sum_{i=p}^{P} \rho_i
\]

\[
T(x) = \frac{x(1 - \rho) + W_0}{(1 - \rho)^2}
\]

\[
\lim_{x \to \infty} T(x) = \frac{x}{1 - \rho}
\]

\[
W(x) = \frac{\rho x}{1 - \rho}
\]

\[
(4.17)
\]

\[
(4.18)
\]
Response Time Distribution in Processor Sharing System: Properties (1/2)

1. Response time is a “linear” function of service time.
2. The mean response time is independent of the service time distribution but only the mean (1st moment).

\[ W(x) = \frac{\rho x}{1 - \rho} \quad (4.18) \]

\[ W = \int_0^\infty W(x)b(x)dx = \rho \bar{x}/(1 - \rho) \]

- It is finite so long as \( \rho < 1 \) and \( x \) is finite
- FCFS, \( x^2 -> \text{infinite} \) -> unbounded delay
Response Time Distribution in Processor Sharing System: Properties (2/2)

- The ratio of waiting time to service time (normalized waiting time)

\[ \frac{W(x)}{x} = \frac{\rho}{1 - \rho} \]

- Consider the equation as a form of penalty function
  - Independent of service time.

3. “fair”
  - Compared to systems where customers with longer service requirements are penalized more heavily per unit of service time.
Response Time Distribution in Processor Sharing System: Properties (3/3)

4. For exponential service time, if having average service time, the response time is the same for both RR and batch processing.

- When \( x = \frac{x^2}{2\bar{x}} \)
- The service time less than average -> shorter response time in processor sharing
- The service time greater than average -> shorter response time in FCFS.

\[
W(x) = \frac{\rho x}{1 - \rho} \quad (4.18)
\]

\[
T(x) = \frac{W_0}{1 - \rho} + x \quad (4.13) \quad \text{where} \quad W_0 = \lambda \frac{x^2}{2}
\]
Finite population model
Introduction

- Previously, we assume the input population is infinite and that the arrival process is Poisson.
- In real-world, the input population is finite.
- When we may approximate an input population by an infinite one?
  - The nature of the arrival process depends only in a negligible way on the number of customers already in the system.
Finite Input Population: example

- Population: $M$ users
- A time-sharing computer
- Station (console) – alternate between the states of “thinking time”, “request processing time”
Finite Population Model

- Thinking time – exponential distribution with mean \( 1/\lambda \) sec.
- If \( M \to \infty \) and \( \lambda \to 0 \) such as \( M \lambda = \lambda' \) (a constant), then we create a Poisson arrival process at average rate \( \lambda' \) from this finite population.
- Assume processor sharing (i.e. RR scheduling)
- Service time – exponential distribution with mean \( 1/\mu \) sec
- FCFS
- \( \to M/M/1/M \)
Finite Population Model (cont’d)

- For M/M/1/M

\[ T = \frac{M / \mu}{1 - P_0} - \frac{1}{\lambda} \] (4.65)

- \( P_0 = \text{prob[no customers in the system]} \)

- Note that the minimum possible response time occurs when \( M=1 \) and \( T=1/\mu \).

- Normalize average system time with the minimum response time

\[ \mu T = \frac{M}{1 - P_0} - \frac{\mu}{\lambda} \]

- \( P_0 \) approaches zero when the number of active stations increases to a “large enough” value.
Observations

- When M is small – a customer is in service while the other customers are in thinking mode.
- When M is large – linear with a slope of unity ($p_0$ approaches zero when the number of active stations increases to a “large enough” value.)
- $M^*$ - saturation point
Is it possible to “saturate” the system?

- Define a notion of **saturation point** as the point where the system becomes “unstable” — meaning infinite average response time.

- We know the system is never saturated for \( \lambda / \mu < 1 \)

- Define another notion of **saturation number** as the ratio of cycle time (sum of avg. thinking time and avg. service time) and avg. service time

\[
M^* \triangleq \frac{1}{\mu} + \frac{1}{\lambda} = \frac{\mu + \lambda}{\lambda} = 1 + \frac{\mu}{\lambda}
\]

(4.66)
Saturation Point

- For a deterministic system
  - all customers require exactly $1/\mu$ sec of processing
    and exactly $1/\lambda$ sec of thinking time
  - We have $M^*$ - the maximum number of customers that the system has no mutual interference between them.

- Each user beyond $M^*$ would cause all other users to be delayed by a time equal to his entire processing time
Saturation Point (cont’d)

- Normalized response time when $M = M^*$
  \[ \mu T = M - \frac{\mu}{\lambda} = (1 + \frac{\mu}{\lambda}) - \frac{\mu}{\lambda} = 1 \]

- For $M >> M^*$, normalized response time is $M - M^* + 1$ times greater than it would be if each user had an entire processor to himself.
  \[ \mu T = M - \frac{\mu}{\lambda} \]

- The RR scheduling (processor sharing) gives linear behavior w.r.t. service time.
Homework

- Pages 253-260: 4.3, 4.10, 4.24 and 4.26