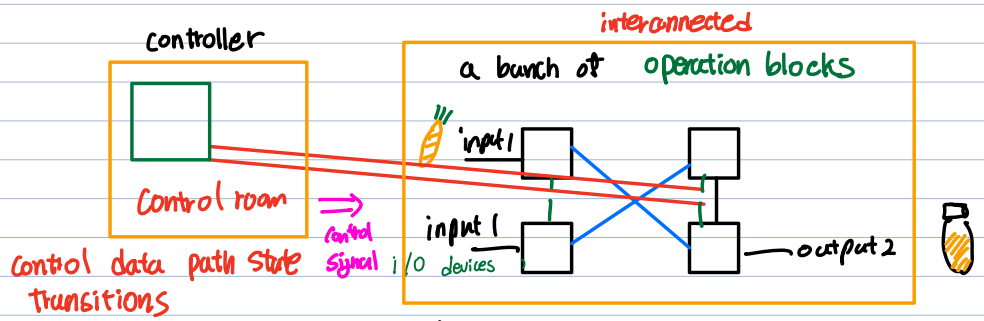
Chapter 2

Two Design System
데이터 시스템은 2개

- Data Flow 데이터 Flow
- Control Flow 컨트롤 Flow

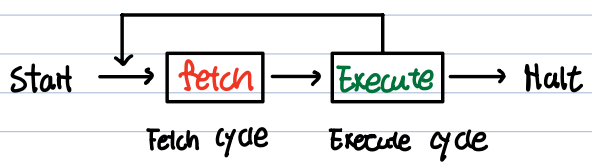
Building blocks
- Any system : beginning and end



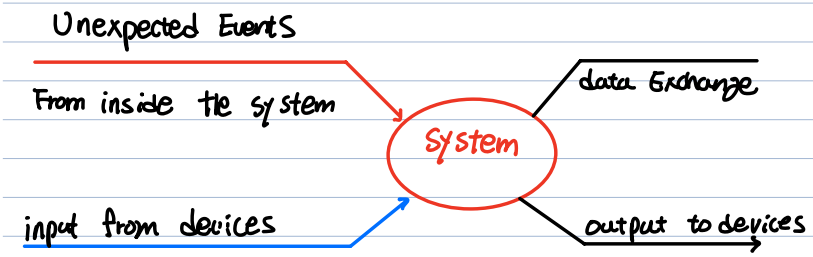
Any operation will take some time

Data path (abstraction How you see set of resources data plane)

A simple controller : Cycle through a sequence of states, in each state, send controls to data path.



STuff Outside of the Control S?



Synchronous : 자발적인 intervention

Exception : division by zero, misaligned instruction
Memory fault : wrong permission , wrong memory translation
sys call : open file , create process

Asynchronous : 비자발적 인터럽 intervention

Timer fired : quantum expiration, packet send timeout : setup alarm
I/O Event: packet received at the NIC (Network interface controller), rendering completed at GPU
Hardware fault: CPU overheating , memory corruption detected Control C
↓ interrupt from other system

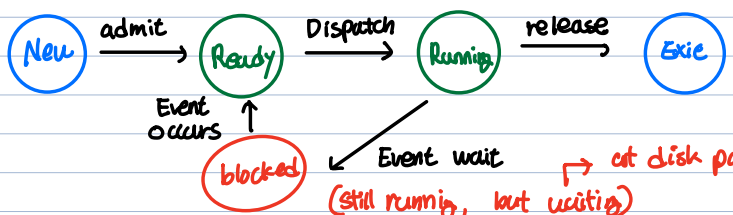
Process : Program in execution on a system , one of many

"State" is necessary
memory stored in , text, Heap, Stack

Process
1. [Consume some resources] (e.g. CPU)
2. Request / wait for an external service / event
3. Do nothing until external service done (wait) : Asynchronous
4. Back up!

Ready
Running
Blocked

Transition State Diagram system is the state of process.

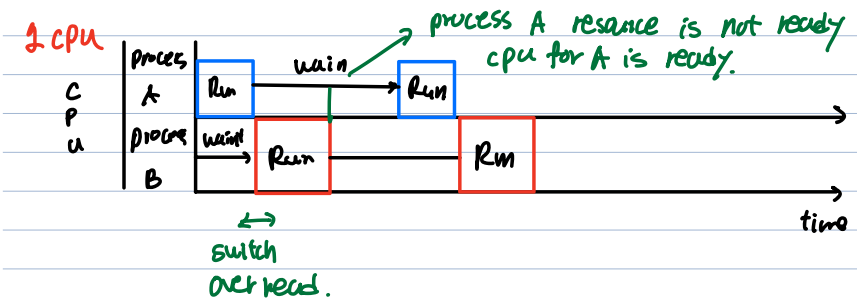


↳ at disk part of program for that resource
(still running, but waiting)
↳ if the process then wants a different resources

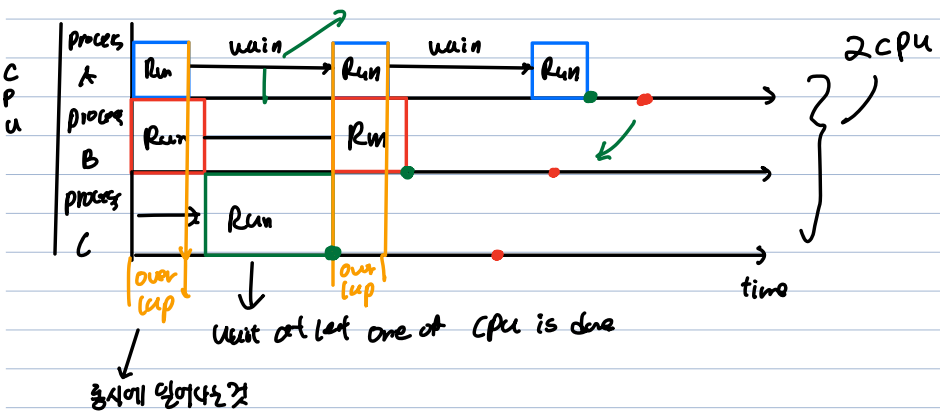
Chapter 3: Processes as Resource Consumers

3.1 concurrency in computing system

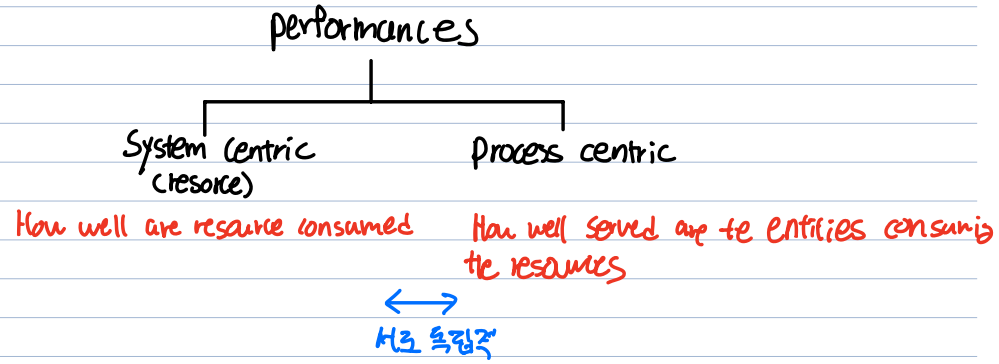
Multi programming (multi-tasking)
: Multiple programs must be executed on the same resource 내가 보통공부 할때들



Multi processing
Overlapped concurrent execution

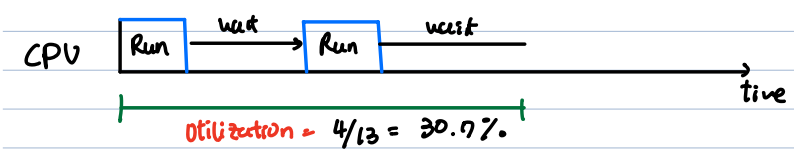


Chapter 4: Performance Metrics and Perspectives



System-Centric performance Metrics server perspective

1 Utilization : a system resource S. over a period of Time T



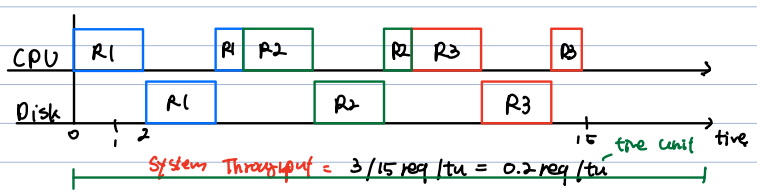
$$U = \frac{S}{T}$$

Max: 1 (100%)

$$(U = 1 - \frac{I}{T})$$

Utilization is a resource specific metric → Utilization 은 각 resource 마다 존재

2. Throughput : Number of completed requests over a given time window
To measure "holistic" performance of a system



$$H = \frac{N}{T}$$

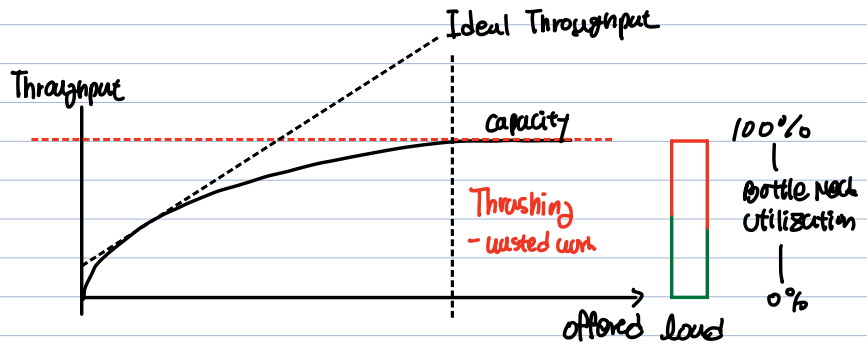
N: # of process

Note: ~~for single resources~~, or for the whole system!

Increasing the MPL improves Utilization and throughput!
But ten is this forever?

3. Capacity

Throughput은 request 증가에 따라 영향을 받아서 (0에 도달할 수) 모델을 정확히 평가 하기가 어렵다.
→ Throughput이 낮다고 다 안좋은 모델이 아니다.



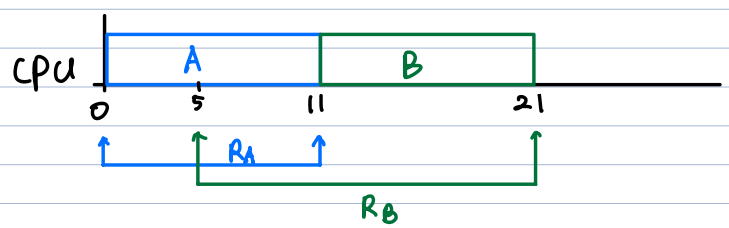
Demand가 늘어나면서 Throughput이 늘어가는게 ideal, 하지만 Bottleneck을 지나면 더이상 늘 이상할 수 있다. (Utilization이 100%가 아니어도, 더 이상 늘어날 수 없으면 bottle neck)
Culprit: when throughput reaches maximum

Process-centric Performance Metrics

n service requests. submitted t_1, t_2, t_3, finished t_1', t_2', t_3' respectively

1. Response Time

$R_i = t_i' - t_i$



R_A: 11 - 0 = 11

KB : 21 - 5 = 16

2) Jitter (Response Time Variability) . Variance, Standard deviation

Average Response Time ≠ "real" response Time

Exponential, normal, uniform, 여러가지 distribution이 있겠지

3. Gaurantee Ratio

Ex). if a deadline is imposed on the service requests, then the percentage of requests that end up meeting their deadlines is the gaurantee ratio.

4.) Lag and Lateness
" How much later after the deadline service was completed : 얼마나 time unit

5) Fairness
requested execution times C_1, C_2, C_3

response time: R_1, R_2, R_3 fair system will keep the ratios R_i/C_i

6.) Speed up and Slow down
System upgrade was beneficial or detrimental.

- Performance expressed in Time before the change of interest : T_{old}. ex) Average response Time
- Same matrics change under analysis : T_{new}

Speed UP : How many times (X) faster is the new system compared to the old one

$$\text{Speed up} = \frac{T_{old}}{T_{new}} \rightarrow \underline{\text{Shorter}}$$

Slow down : Inverse of the speed up and returns a value greater than 1 indicates a performe loss

$$\text{Slowdown} = \frac{T_{new}}{T_{old}}$$ 1 보다 작아야 좋다.

Amdahl'S Law : overall performance improvements of the system.

f: fraction of time a resource is used in serving request
X: improving the performance of that resource x-fold.
Y: overall performance of the system by a factor ← speed up $\frac{T_{old}}{T_{new}}$

$$Y = \frac{1}{1 - f \cdot (1 - \frac{1}{x})}$$

f: parallelize proportion
x: number of process

Other Aspects of "System Performance"

Real system, failures happens!

1. Reliability : Process-centric (user 관점)

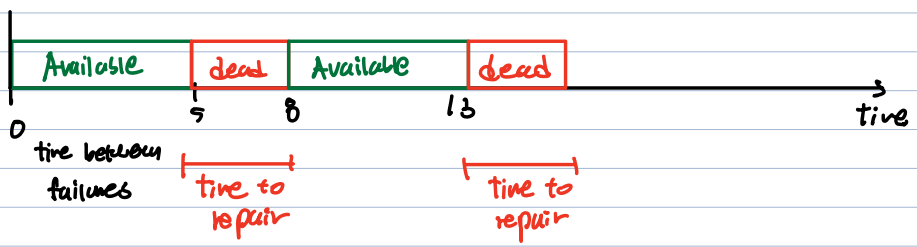
measure of the probability that the system functions correctly continously for a specificed time period. → system 시간 관점

$$R(t) = e^{-t/MTBF}$$

2) Availability : Process-centric (client 관점)

on-line solution is a measure of the probability that a system is available at an arbitrary point in time (→ system 동작하는 관점)

→ 99% availiability, 99% reliability
1000 request 중 10개 refuse 되고 990 request 중 3%가 failure 일 것이다.



$$\text{Availability} = \frac{\text{up Time}}{\text{Total time}} = \frac{\text{mean Time between failures (Available)}}{\text{mean Time between failures} + \text{Mean Time to Repair (dead)}}$$

$$\simeq \frac{MTBF}{MTBF + MTTR}$$

MTBF : $\frac{\text{Up Time (available total)}}{\text{N fail}}$

MTTR = $\frac{\text{down Time (System down total)}}{\text{N fail}}$

} maintainability

Chapter 5 Performance Evaluation and Models

How done obtain (utilization, response time, jitter, etc) from a given system?
시스템 설계자들이 시스템을 모델링하고 분석함으로써 자동 사용하는 도구들 " Queing Analysis"

Queing Analysis

Customers: Individual requests for services
Queues: waiting areas were requests for service waits.
Servers: satisfying service requests

Distributions of Arrivals: Poisson process for the arrival of customers. The rate of arrivals λ
# of customers coming into the system every period T is $\lambda \cdot T$ / Arrival of requests are independent

Distribution of Service Time: Mean service Time: T_S 서버가 처리하는데 걸리는 시간
μ (service rate) : 단위 시간당 서버가 처리할 수 있는 요청의 수 T_S와 반비례

$$\mu = \frac{1}{T_S}$$

Ex) 평균 서비스 시간이 2초이면, 서버는 평균적으로 1초당 0.5개의 요청을 처리할 수 있습니다. ($\mu = 1/2 = 0.5$)

1) Single Queue, Single-Server System

Queue Size: Unlimited.
Population: independent.

1) W : # of customers waiting in the line

2) q : Total # of customers in the system

3) T_s : Service Time in Server

3) T_s : service time in server

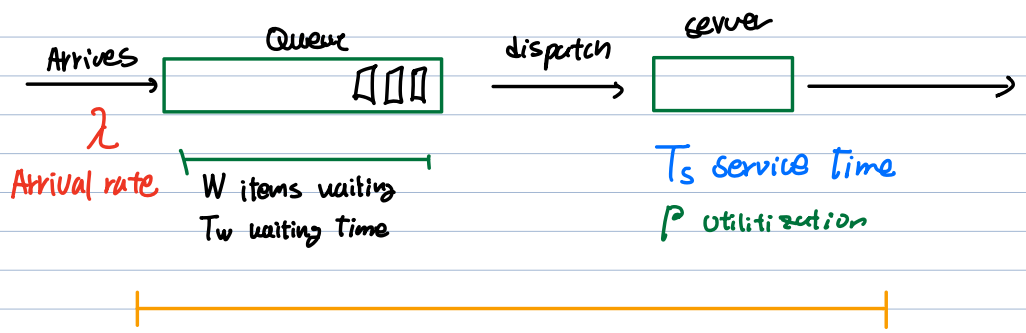
4) T_w : waiting Time in Queue

5) T_q : turnaround Time [response Time] req/sec × sec/req = 노란색

6) ρ : Utilization of the system $\rho = \lambda \cdot T_s = \frac{\lambda}{\mu}$ 들 단위가 같아.

In steady state, the rate at which cannot exceed the rate at which the server is able to serve them.

$$\lambda < \mu \Rightarrow \rho < 1$$



Little's Law

True for any "steady-state" queing system.

$q = \lambda \cdot T_q$ and $w = \lambda \cdot T_w$

Relationships of Little's Law

$T_q = T_s + T_w$

$q = w + \lambda \cdot T_s = w + \rho = w + \frac{\lambda}{\mu}$

$W = \lambda \cdot T_w$

$q = \lambda \cdot (T_w + T_s) = \lambda \cdot T_q$

- Request to a web server experiences average response time of 0.5 seconds and that the average rate of arrivals for requests is 100 requests per second., the average web server would be manging 50

$T_q = 0.5$ sec/req $\quad \lambda = 100$ req/s $\quad q = 100 \times 0.5 = 50$ req

- Multiprogramming Level (number of process in system) is 30 on average and that the ratio with which processes are created is 2 per minutes. then you may conclude that the average "life time" of a process is $30/2 = 15$

$q = 30 \quad \lambda = 2 \quad T_q = 30/2 = 15$ req/min

- Rate of arrival of packets to a router is 1000 per second and that it takes 0.5 msec to service each one of these packets and that the average number of packets queued at the router is 3, then "delay through this router" will be

$\lambda = 1000$ req/sec $\quad T_s = 0.5$ msec/req $= 0.0005$ sec/req

$W = 3$ req

T_q

$T_q = T_w + T_s \quad T_w = \frac{W}{\lambda} = 3 \times \frac{1}{1000} = \frac{3}{1000} = 0.003$ sec/req

$T_s = 0.0005$ sec/req

$T_q = 0.003 + 0.0005 = 0.0035$ sec/req $= 3.5$ msec/req

$\lambda = 100$ req/sec

Chapter 7 : Basic Probability Analysis

Probability = quantification of (probable)

Modeling Probabilistic Outcomes

Probability helps in modeling the "unknown" because outcomes are "unknown"

Random variables

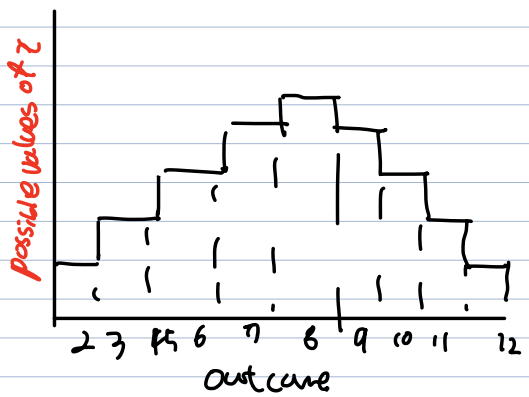
systems examples
- throughput, arrival rate, utilization.
- Response time.

# of processes blocked at a given time
The time interval between two failures — Exponential
Time taken to access a hierarchical memory

Probability distribution

Probability Mass function

$P(X=2)$
$P(X=3)$
$P(X=4)$
$P(X=5)$



How about continuous distribution

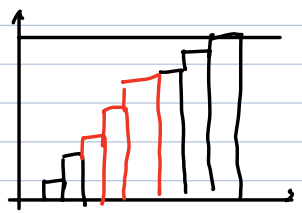
' outcome of experiment '

Cumulitive

Extracting More From F(x)

F(x) to find the probability

$P(x_1 < x \le x_2) =$



$= P(x \le x_2) - P(x \le x_1)$

$= F(x_2) - F(x_1)$

$= \int_{-\infty}^{x_2} f(x)\,dx - \int_{-\infty}^{x_1} f(x)\,dx$

$= \int_{x_1}^{x_2} f(x)\,dx$

Chapter 8 : Probability Distributions as Modeling Tools

1. Uniform distribution

$f(x) = \begin{cases} c & \text{if } x \ge a \text{ and } x \le b \\ 0 & \text{if } x < a \text{ or } x > b \end{cases}$ - a, b 사이

$\int_{-\infty}^{\infty} f(x)\,dx = 1$

$\int_{-\infty}^{\infty} f(x)\cdot dx = \int_a^b c\cdot dx = c\cdot b - ca = c(b-a) = 1$

$\therefore c = \frac{1}{b-a}$

Mean $\mu = \frac{b+a}{2}$ Variance $\sigma^2 = \frac{(b-a)^2}{12}$ $\sigma = \sqrt{\frac{(b-a)^2}{12}}$

2) Bernoulli distribution

Coin, a random trial with binary output

$f(x) = \begin{cases} p & \text{if } x = 0 \text{ fail} \\ 1-p & \text{if } x = 1 \end{cases}$) 1-p 가장 흔한 곳들 Geometric 에서 쓰일거야

$\mu = 0\cdot p + 1\cdot(1-p) = 1-p$ $\sigma^2 = E[(x-\mu)^2] = E[x^2] - \mu^2$

$$0^2 \cdot p + 1^2(1-p) - (1-p)^2$$

$$= 1 - p - (1 - 2p + p^2)$$

$$= 1 - p - 1 + 2p - p^2$$

$$= p - p^2 = p(1-p)$$

3) Geometric distribution

PMF $f(n) = p^n(1-p)$, $0 \le p \le 1$, $n \ge 0$

성공전 n 번의 실패

mean: $\mu = \sum_{n=0}^{\infty} n \cdot p^n(1-p) = \frac{p}{1-p}$ converge

$\frac{1}{0.9} = \frac{10}{9} = 1.11111$

하지만 평균적으로 첫성공을 하기까지의 시행 횟수는

$f(m) = f(n-1) = p^{n-1} \cdot (1-p)$ $\mu = \frac{1}{1-p}$

$\sigma^2 = \frac{p}{(1-p)^2}$

4. Binomial Distribution

Conduct Bernoulli trial n times

$f(x) = \binom{n}{x} \cdot p^x \cdot (1-p)^{n-x}$

$\binom{n}{x} = \frac{n!}{(n-x)!\,x!}$

mean: $\mu = np$ Varience: $n \cdot p(1-p)$

The probability that a sever is "up" is 0.9 what is the probability that i severs will be up?

$p = 0.9$

$n = 3$

$f(0) = \binom{3}{0}(0.1)^0(0.3)^3 = 1 \cdot 1 \cdot (0.3) = 0.027$

$f(1) = \binom{3}{1}(0.1)^1(0.3)^2 = \frac{3!}{1!2!} = \frac{3\times2\times1}{2\times1} = 3 \times 0.1 \times 0.09 = 0.189$

$f(2) = \binom{3}{2}(0.1)^2(0.3)^1 = \frac{3!}{2!1!} = \frac{3\times2\times1}{2\times1\times1} \times (0.1)^2 \cdot 0.3 = 0.441$

$f(3) = \binom{3}{3}(0.1)^3 = \frac{3\times2\times1}{3\times2\times1} = (0.1)^3 = 0.343$

5 Poisson Distribution

Consider binomial Process with large n and a small p. $n \to \infty$, $p \to 0$. Product $np = \lambda > 0$.

λ : 단위 시간 또는 단위 공간에서 특정 사건이 발생하는 평균 발생 횟수

a) 1시간당 평균적 5번의 전화 : $\lambda = 5$ / 1분에 10명의 방문자 : $\lambda = 10$

$$\text{PMF } f(x) = \frac{\lambda^x}{x!} e^{-\lambda}$$

x는 사건이 발생한 횟수

$\mu = \sigma^2 = \lambda$

$$f(x) = \lim_{n\to\infty,\, p\to 0} \binom{n}{x} p^x (1-p)^{n-x} = \frac{\lambda^x}{x!} e^{-\lambda}$$

CMF 다 외워라.

ex) $P(2)$ = 2번 일어날 확률
$P(3)$ = 3번 일어날 확률

6. Exponential distribution

사건이 발생할 두 시간 사이의 시간 간격의 모형임

λ : 단위시간당 사건이 발생할 것으로 기대되는 평균 발생 횟수

$$f(x) = \lambda e^{-\lambda x} \quad x > 0$$

사건이 한 시간에 5번 발생할 때 사건 사이의 평균 시간 간격은 $\frac{1}{5}$ 시간

$F(x) = 1 - e^{-\lambda x}$

$\mu = \frac{1}{\lambda}$ $\sigma^2 = \frac{1}{\lambda}$

A Note on Reliability

If it is better than fails every MTBF

then its failure rate is 1/MTBF

... hence probability of failure at t or earlier

$F(t) = 1 - e^{-t \times MTBF}$

... hence prob of being "up" for t or longer

$1 - F(t) = e^{-t \times MTBF}$

→ MTBF가 5분이라고 했을때

$\lambda = \frac{1}{5}$

↓

1분에 failure 할 확률 5분

↓

failure 하는 두 시간 사이의 간격이 MTBF (Mean Time between failure

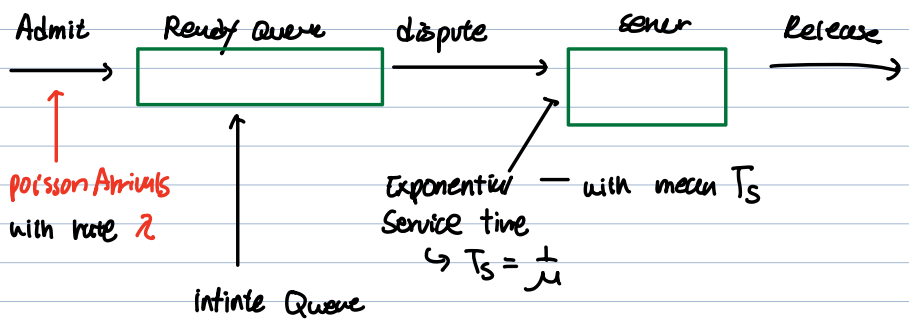
Chapter 9 Elementary Queing Analysis

Analysis of M/M/1 Queing system

Single queuing single server

Arrival rate is poisson and service time is exponential

M/M/1 System



Consider a very small interval of time of length h. Assume that this interval time (h) is so small that a max of one arrival can realistically occur in that period of time. Since the rate of arrival is λ per request/unit tie rate of arrival per interval is λh. if λ = 100req/sec and h = 1msec rate of arrival = 0.001, 0.1 req/msec.

Probability density of the poisson distribution is

$$f(x) = \frac{\lambda^x}{x!} e^{-\lambda}$$

given that the rate of arrival per interval h is λh, the probability of x arrivals per interval h is

$$f(x) = \frac{(\lambda h)^x}{x!} e^{-(\lambda h)}$$

$f(0)$ = no arrival is $f(0) = \frac{(\lambda h)^0}{0!} e^{-(\lambda h)} = e^{-\lambda h}$

so $f(\text{no arrival}) = e^{-\lambda h}$

Let's say arrival to the system is birth

$$P(\text{birth}) = 1 - P(\text{no arrival}) = 1 - e^{-\lambda h}$$

Rewrite using taylor series

$$1 - e^{-\lambda h} = 1 - \left(1 - \lambda h + \frac{(\lambda h)^2}{2!} - \frac{(\lambda h)^3}{3!} + \cdots\right)$$

1 or more arrival

so $P(\text{birth}) = \lambda h$

Similar to this, finish request is "Death"

rate of finish service is μ req/sec. so for the interval of time h, the rate of finish service is μh.

so I can use poisson distribution

$$f(y) = \frac{(\mu h)^y}{y!} e^{-\mu h}$$

when $y = 0$ $f(0)$: For a given time h, probability of finishing request $= 0$. so the probability of finishing request, "death" is

$$\frac{(\mu h)^0}{} - \mu h$$

$$1 - f(0) = 1 - \frac{(\mu h)^0}{0!} e^{-\mu h} = 1 - e^{-\mu h} = \mu h$$

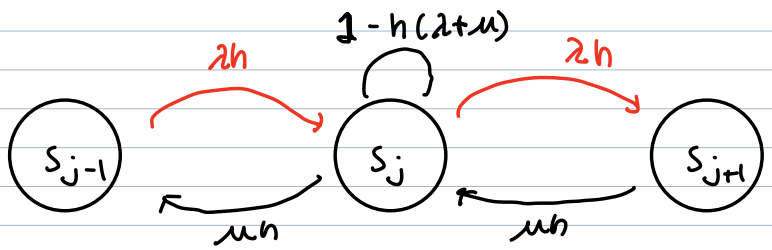
$P(\text{death}) = \mu h$

1 or more request finished

State Transition diagram for M/M/1

State of the system by S_j j customers are in the system

1. The system was in S_{j-1} and a birth occured, the probability of that happening is λh
2. The system was in S_{j+1} and a death occured, the probability of that happening is μh
3. The system was in state S_j and, neither death or birth occured. The probability of that happening is $1 - \lambda h - \mu h = 1 - h(\lambda + \mu)$



Relationship: $P(S_j) = \lambda h \cdot P(S_{j-1}) + (1 - h(\lambda+\mu)) P(S_j) + \mu h \cdot P(S_{j+1})$

Re arrange

$$0 = \lambda h \cdot P(S_{j-1}) + (1 - h(\lambda+\mu)) P(S_j) - P(S_j) + \mu h \cdot P(S_{j+1})$$

$$0 = \lambda h (P_{S_{j-1}}) - h\lambda P(S_j) - \mu h P(S_j) + \mu h P(S_{j+1})$$

$$0 = \lambda P(S_{j-1}) - \lambda P(S_j) - \mu P(S_j) + \mu P(S_{j+1})$$

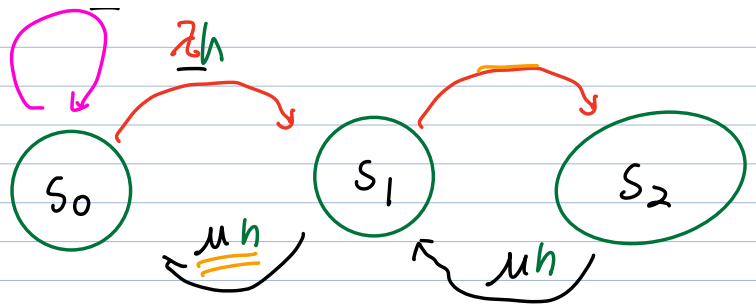
$$(\lambda + \mu) P(S_j) - \lambda P(S_{j-1}) = \mu P(S_{j+1})$$

$$P(S_{j+1}) = (\rho + 1) P(S_j) - \rho P(S_{j-1})$$

$(\rho+1)$: 1 + util; ρ: − util

S_0 : system state in which there are no requests

$1 - \lambda h$



$$P(S_0) = (1-\lambda h)P(S_0) + \mu h P(S_1)$$

$$\lambda P(S_0) = \mu P(S_1)$$

$$\left(\frac{\lambda}{\mu}\right) = \rho$$

$$\rho P(S_0) = P(S_1)$$

$$\boxed{P(S_1) = \rho P(S_0)}$$

$P(S_2)$ = $j=1$

$$P(S_2) = (1+\rho)P(S_1) - P(S_0)$$
$$= (1+\rho)\cdot \rho P(S_0) - P(S_0)$$
$$= \rho^2 P(S_0)$$

$$P(S_3) = P(S_3) = (1+\rho)P(S_2) - \rho P(S_1)$$
$$= (1+\rho)\rho^2 P(S_0) - \rho\cdot\rho P(S_0)$$
$$= (\rho^2+\rho^3-\rho^2)P_{S_0}$$
$$= \rho^3 P_{S_0}$$

$\therefore$ so $P(S_j) = \rho^j P(S_0)$

overall probability는 1이되어야 하므로

$$\sum_{n=0}^{\infty} P(S_j) = \sum_{n=0}^{\infty} \rho^i \cdot P(S_0) = 1$$

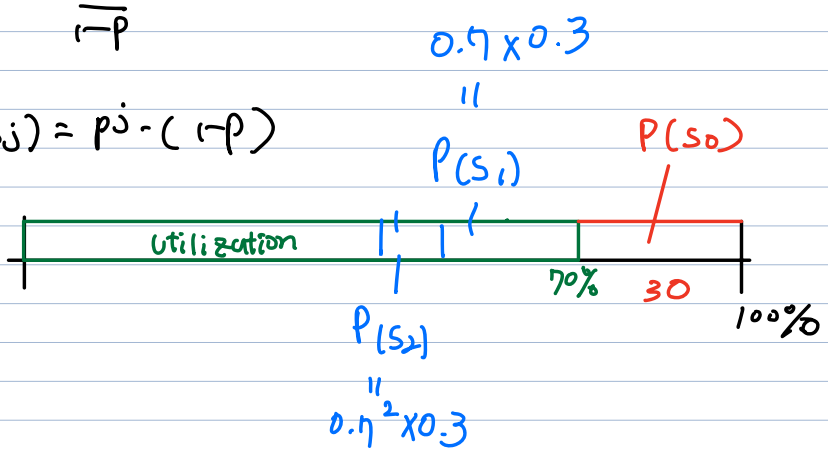
- $P(S_0) = \frac{1}{\sum_{i=0}^{\infty} p^i}$

$\sum_{i=0}^{\infty} p^i$ 는 Geometric series. p는 1보다 작은 값 (probability)

$\mu = \frac{1}{1-p}$ 일 때 converge.

$P(S_0) = \frac{1}{\frac{1}{1-p}} = 1-p$

$P(S_j) = p^j \cdot (1-p)$



Average number of customer in an M/M/1 System

How can we compute average number of customers q in an M/M/1 system.

$$q = E[\text{number of customers}] = \sum_{n=0}^{\infty} i \cdot P(S_i) = \sum_{i=0}^{\infty} i \cdot p^i \cdot (1-p)$$

$$= (1-p) \sum_{i=0}^{\infty} i \cdot p^i$$

$$q = (1-p) \sum_{i=0}^{\infty} \frac{p^i}{i^{-1}}$$

↓ polylogarithms

$$= (1-p)\, Li_{-1}(p)$$

$$Li_{-1}(p) = \frac{p}{(1-p)^2}$$

$$= (1-p) \times \frac{p}{(1-p)^2}$$

$$q = \frac{\rho}{1-\rho}$$

Average number of customers waiting for service in a M/M/1 system

$$w = q - \rho = \frac{\rho}{1-\rho} - \rho = \frac{\rho - \rho + \rho^2}{1-\rho} = \frac{\rho^2}{1-\rho}$$

Average time in an M/M/1 system

$$T_q = \frac{q}{\lambda} = \frac{\rho}{1-\rho} \div \lambda = \frac{\rho}{1-\rho} \times \frac{1}{\lambda} = \frac{\frac{\lambda}{\mu}}{1-\rho} \times \frac{1}{\lambda} = \frac{\frac{1}{\mu}}{1-\rho} = \frac{1}{\mu(1-\rho)}$$

Average Time waiting in the M/M/1 system

$$T_w = \frac{w}{\lambda} = \frac{\rho^2}{1-\rho} = \frac{\rho}{\mu(1-\rho)}$$

Slow down

Slow down is defined as average length of time it takes to get rendered (response time) over minimum amount of time in which can be rendered (service time)

$$\text{Slow down} = \frac{T_q}{T_s} = \frac{\frac{1}{\mu(1-\rho)}}{\frac{1}{\mu}} = \frac{1}{\mu(1-\rho)} \times \frac{\mu}{1} = \frac{1}{1-\rho}$$

Variability measures

Standard deviation in the number of customers in an M/M/1 system as well as the total time in the system.

$$\sigma_q = \frac{\sqrt{\rho}}{1-\rho}$$

$$\sigma_{T_q} = \frac{1}{\mu(1-\rho)} = \frac{T_s}{1-\rho}$$

Ex) M/M/1 Example

1. Poisson Arrival rate $\lambda = 10$ req/sec, service exponential $T_s = 9$ msec/req

Calculate response Time T_q

$T_s = 0.009$ sec/req $\quad \lambda = 10$ req/sec

$\rho = 0.009 \times 10 = 0.09$

$q = \frac{\rho}{1-\rho} = \frac{0.09}{0.91} = 0.098901098$

$T_q = q \div \lambda = 0.009890109 8$ sec/req

2. Due to super bowl, hit to the webserver went up to 10 folds $= \lambda \times 10$

$\lambda = 100$ req/sec

response time $T_q =$

$\rho = 0.009 \times 10 = 0.9$

$q = \frac{0.9}{0.1} = 9$

$T_q = q \div \lambda = q \div 100 = 0.09$

3. relative slow down

Slow down $= \frac{T_q}{T_s}$

Before : slow down : $\frac{0.0099}{0.009} = 1.099$

After the super bowl : 10

4 How much faster the web server be.

$$\frac{(0.099 \times x) \times 100 = \rho}{1 - (0.009 x \times 100)} = \frac{q}{(0.099x} = 1.09$$

$$\frac{qx}{\quad} \times \frac{(}{\quad 0000}$$

$1 - \alpha x$ $\quad \therefore \frac{0.009x}{0.001} = \frac{1}{1}$

$$\frac{1000}{1-9x} = 1.09^{\frac{1}{1000}}$$

$$1.09 - 9.81x = 1000$$

$$x = 101$$

Chapter 11: Variations of the M/M/1 Queue System

Exponential distribution is memory less.

What if service time is not exponential

11.1) Constant time M/D/1

service time is constant. All customers require the same amount of service.

$$q = E[\text{number of customers}] = \frac{\rho^2}{2\cdot(1-\rho)} + \rho \quad \text{Idk why}$$

$$w = q - \rho = \frac{\rho^2}{2(1-\rho)}$$

$$T_q = q \div \lambda = \frac{\rho}{2\cdot\mu(1-\rho)} + T_s \qquad W_q = w \div \lambda = \frac{\rho}{2\cdot\mu(1-\rho)}$$

11.2) General service Time - M/G/1

we do not know the distribution of service time

But 1) Average service Time, T_s, service Time standard deviation σ_{T_s}

2) Normalized value of the service time standard deviation σ_{T_s}/T_s

A $\left| \frac{1}{2}\cdot\left[1 + \left(\frac{\sigma_{T_s}}{T_s}\right)^2\right] \right.$

Comput q and w

$$q = \frac{\rho^2 \cdot A}{1-\rho} + \rho \qquad w = \frac{\rho^2 \cdot A}{1-\rho} \qquad \rightarrow \quad T_q = \frac{\rho \cdot A}{\mu(1-\rho)} + \frac{1}{\mu} = \frac{1}{\mu}\left(\frac{\rho \cdot A}{(1-\rho)} + 1\right) \qquad W_q = \frac{\rho \cdot A}{\mu(1-\rho)}$$

Sanity check

1) M/M/1

For an exponential distribution, Standard is equal to the mean $\sigma_{T_S} = T_S$, $\sigma_{T_S}/T_S = 1$

A for M/M/1 : $\frac{1}{2}[1+(1)^2] = 1$

$$q = \frac{\rho^2}{1-\rho} + \rho = \frac{\rho^2 + \rho - \rho^2}{1-\rho} = \frac{\rho}{1-\rho}$$

↓

q가 같은지 나오는지 확인.

2) M/d/1

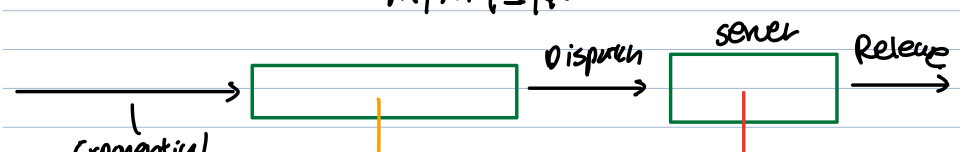
For an deterministic, Standard is 0

A for M/d/1 : $\frac{1}{2}[1+0^2] = \frac{1}{2}$

$$q = \frac{\rho^2 \cdot \frac{1}{2}}{(1-\rho)} + \rho = \frac{\rho^2}{2(1-\rho)} + \rho$$

11.3 Finite Queue Systems - M/M/1/k

M/M/1/k:



Exponential

finite Queue size

Exponential service time

M/M/1/k 에서도 $\rho > 1$도 허용됩니다.

$P(S_j)$는 Q가 j개 의뢰청을 확률 $j=0$일때 아무 요청이 없는상태, $j=k$ 일때는 가득 찬 상태

$$P(S_j) = \begin{cases} \dfrac{(1-\rho)\cdot\rho^j}{1-\rho^{k+1}} & \rho \neq 1 \\ \dfrac{1}{k+1} & \rho = 1 \end{cases}$$

계속 클라이언트가 일정하게 full

$\rho=1$ 최대로 요청을 내고있는게 full

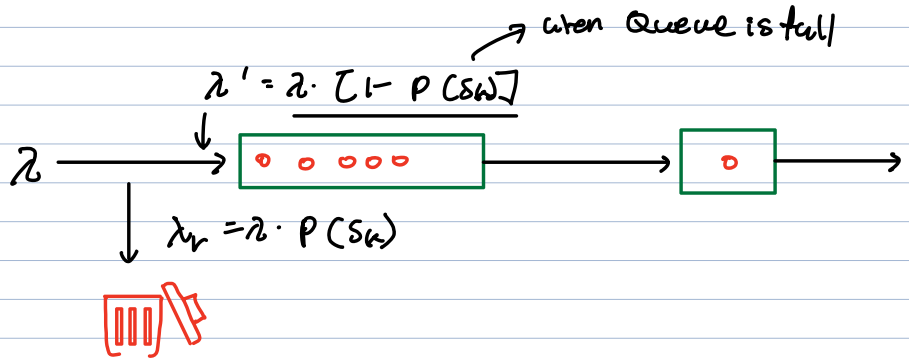
$$q = \begin{cases} \dfrac{\rho}{1-\rho} - \dfrac{(k+1)\cdot\rho^{k+1}}{(1-\rho^{k+1})} & \text{if } \rho \neq 1 \\ \dfrac{k}{2} & \text{if } \rho = 1 \end{cases}$$

M/M/1/k

Rejection Rate

$P(S_k)$ = P(rejection) 거부될 확률 Queue가 k개로 full

$$P(S_k) = \begin{cases} \dfrac{(1-\rho)\rho^k}{1-\rho^{k+1}} & \rho \neq 1 \\ \dfrac{1}{k+1} & \rho = 1 \end{cases}$$



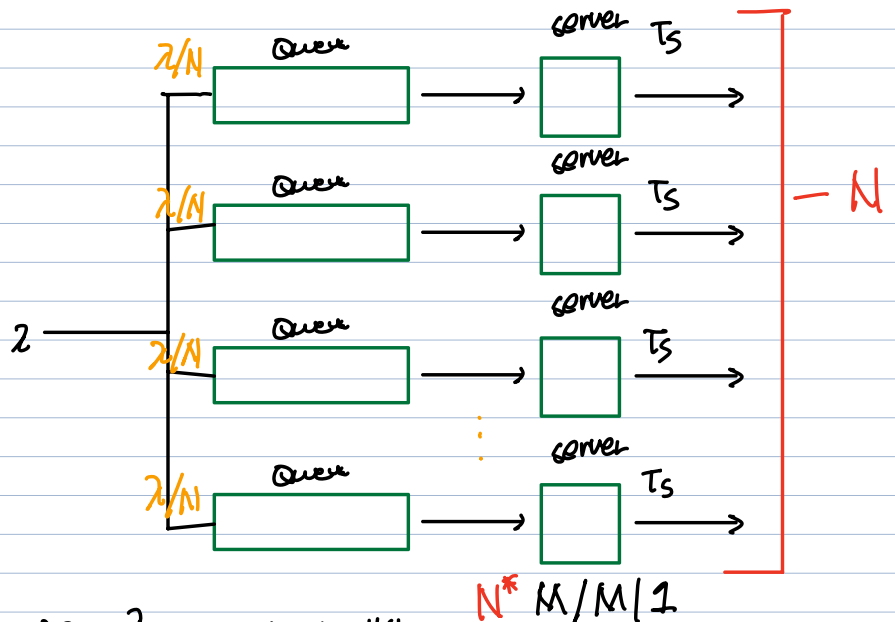
∴ For M/M/1/k $T_q = \dfrac{q}{\lambda'}$

Chapter 12: Networks of Queue

2) Multiple Single-sever Queues

Customers choose one of N Queues. Grocery stores Star market

$N * M/M/1$



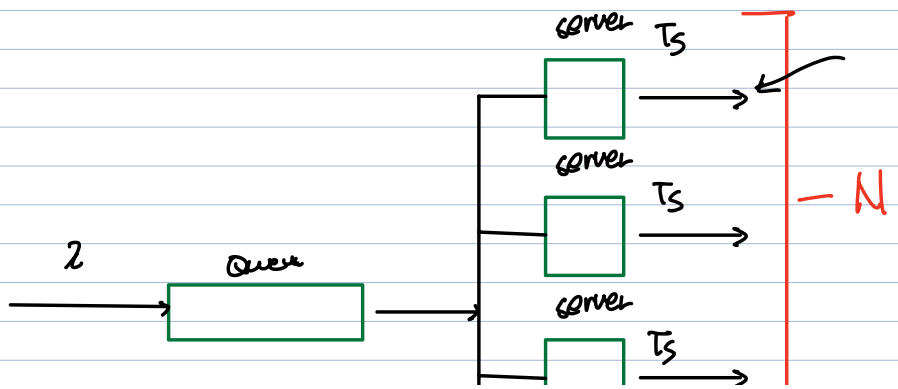
λ를 $\frac{\lambda}{N}$로 나눠서 계산

$N \times q = q_{tot}$ $\quad P' = \frac{\lambda}{N} \cdot T_S$

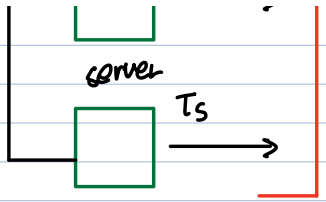
$q' = \frac{P'}{1-P'}$ $\quad N \times q' = q_{total}$

2) Multi-server Queue

$M/M/N$



M/M/N



Probability N or more customers in the system (All servers are busy)

$$C = \frac{1-k}{1-pk}$$

k is a poisson ratio function

$$k = \frac{\sum_{i=0}^{N-1} \frac{(NP)^i}{i!}}{\sum_{i=0}^{N} \frac{(NP)^i}{i!}} = 1 \cdot \frac{\frac{(NP)^N}{N!}}{\sum_{i=0}^{N} \frac{(NP)^i}{i!}}$$

$$\text{erlang-C } P = \frac{\lambda T_S}{N} = \frac{\lambda}{\mu N}$$

Utilization of Single server are equal

$$P' = \frac{\lambda}{N} T_S$$

$$q \mid \frac{P}{1-P} \cdot C + NP \rightarrow \text{Requests being served}$$

$\frac{P}{1-P}$: 한 서버의 Utilization × C: probability of all severses are busy

Example

2 servers

$\lambda = 20$ $\quad T_S = 80\text{msec/req} = 0.08\text{ sec}$

Goal: Compute Response Time T_q

1) N M/M/1 $\Rightarrow$ 2 * M/M/1

$\lambda = 20 / 2 = 10$ req/sec

$\rho = 10 \times 0.08 = 0.8$

$q' = \frac{0.8}{1-0.8} = 4 \qquad q_{total} = 2 \times 4 = 8$

$T_q = 8 \div \lambda = 8 \div 20 = 0.4$

2) M/M/2

$\rho' = 10 \times 0.08 = 0.8$

Get k first

$$k = 1 - \frac{\frac{(2 \times 0.8)^2}{2 \times 1}}{}$$

$i = 0 \quad \frac{(2 \times 0.8)^0}{0!} = 1$

$i = 1 \quad \frac{(2 \times 0.8)^1}{1!} = 1.6$

$i = 2 \quad \frac{(2 \times 0.8)^2}{2 \times 1} = 1.28$

} 3.88

$k = 0.670$

$= 0.67$

$C = \frac{1 - 0.67}{1 - 0.8 \times 0.67} = \frac{0.33}{0.46} = 0.72$

$q = \frac{0.8}{0.2} \times 0.72 + 2 \times 0.8 =$

$2.88 + 1.6 = 4.48$

$q = 4.48$

$T_q = 0.224$

$$q = \frac{r}{1-p} \cdot C + Np$$

Average Number of requests waiting for service in M/M/N

$$w = q - Np$$ → 현재 시계의 버퍼로에 들어가있음.

$$= \frac{p}{1-p} \cdot C + Np - Np = \frac{p}{1-p} \cdot C$$

Average Time in an M/M/N System

$$T_q = \frac{q}{\lambda} = \frac{Np'}{\lambda} + \frac{p'}{\lambda(1-p)} \cdot C$$

$$p' = \frac{\lambda}{N} \cdot T_s \qquad = \frac{N T_s}{N} + \frac{T_s}{N(1-p)} \cdot C$$

$$= T_s + \frac{T_s}{N(1-p)} C$$

$$\sigma_{T_q} = \frac{T_s}{N(1-p)} \sqrt{C(2-C) + N^2(1-p)^2}$$

Average waiting Time M/M/N

$$T_w = \frac{w}{\lambda} = \frac{p}{1-p} \cdot C \times \frac{1}{\lambda} = \frac{T_s}{N(1-p)} \cdot C$$

3) General Network of Queues

Considering the system at steady state:

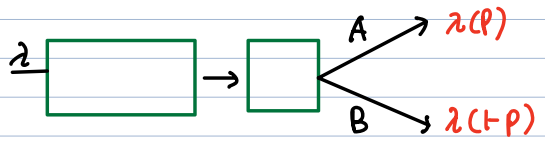
1) Tie rate in must be equal to the rate out of every queue

2) If the arrival process to a queue is poisson, the departure process is also poisson

Steady state arrival rate should be lower than the lowest service rate

utilization $\leq 100\%$, then there will be steady state

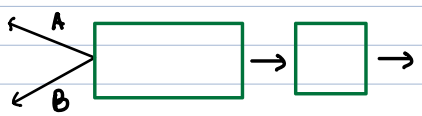
Splitting



1) Path A takes with probability: p
   Path B taken with probability: 1-p

2) If the arrival process is Poisson, then a fixed "portion" p of arrivals is also poisson.

Merging



1) If two flows are joined, resulting flow has sum of the rates

2) If Arrival processes are poisson, with λ_A and λ_B, resulting process also poisson with rate $\lambda_A + \lambda_B$

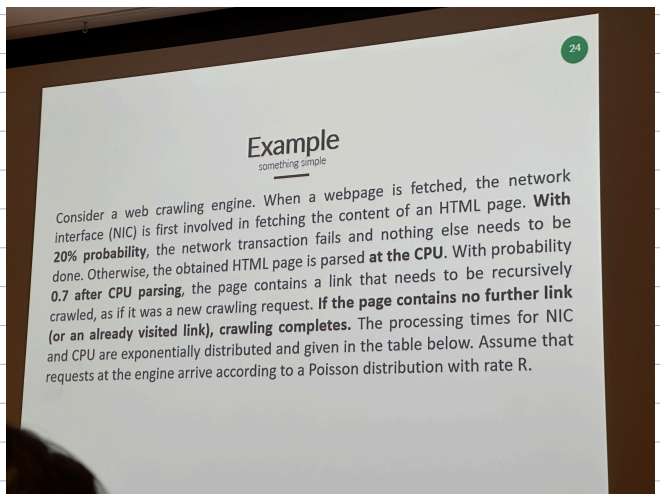
Jackson Theorem

Assumption:

- Steady state
- Arrivals from outside are poisson

- service times for all servers Exponential
- No finite queues ✓

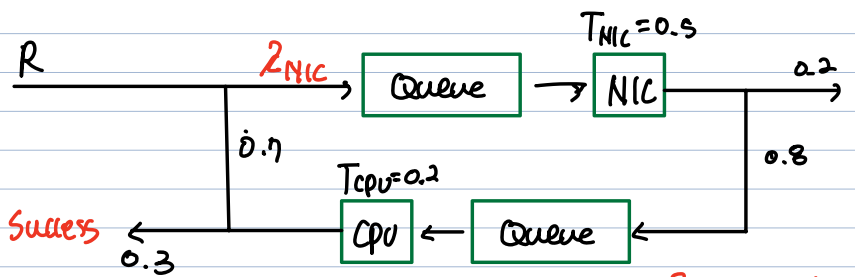
Example slides



Given

	Request processing times
CPU	0.2 sec
NIC	0.5 sec

Assume
- Steady state
- infinite Queue



Geometric Distribution
↓
Success + Fail

λ_{NIC} in has to be the same as λ_{NIC} coming out

✓

$$\lambda_{NIC} = R + 0.8 \times 0.7 \times \lambda_{NIC}$$

$$\lambda_{NIC} = R + 0.56\lambda_{NIC}$$

$$x = R + 0.56x \qquad 0.44x = R$$

$$x = \frac{R}{0.44} = R \times \frac{100}{44} = \frac{50}{22} = \frac{25}{11} R$$

$$\lambda_{NIC} = \frac{25}{11} R$$

전체 시스템

$$\lambda_{NIC} = R + 0.8 \times 0.7 \times \lambda_{NIC}$$

$$\lambda_{NIC} = R + 0.56\lambda_{NIC}$$

$$R = \lambda_{NIC} \cdot 0.2 + \lambda_{NIC} \cdot 0.8 \cdot 0.3$$

$$\lambda_{NIC} = \frac{R}{0.2 + 0.8 \cdot 0.3} = \frac{R}{0.44}$$

$$P_{NIC} = \lambda_{NIC} \cdot T_{NIC} = \frac{R}{0.44} \cdot 0.5 =$$

$$\lambda_{CPU} = \lambda_{NIC} \cdot 0.8$$

$$P_{CPU} = \lambda_{CPU} \cdot T_{CPU} = \frac{0.8R}{0.44} \cdot 0.2$$

$$q_{NIC} = \frac{P_{NIC}}{1 - P_{NIC}}$$

$$q_{CPU} = \frac{P_{CPU}}{1 - P_{CPU}}$$

$$\boxed{q_{Tot} = q_{NIC} + q_{CPU}} \qquad T_{q\,tot} = \frac{q_{TOT}}{R}$$

$$\frac{W_{total}}{R} = T_{W\,total} \qquad \frac{T_{q\,total}}{T_{w\,total}} = T_{S\,total}$$

$$T_q = T_S + T_W \qquad \frac{T_{q\,total}}{T_{s\,total}} = \text{slowdown}$$