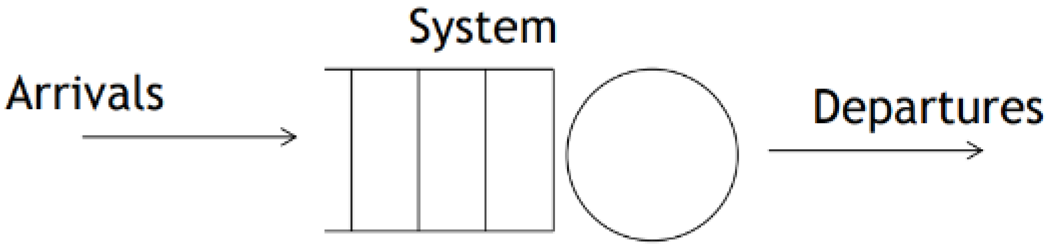
Operational Laws

Operational laws are simple equations which may be used as an abstract representation or model of the average behaviour of almost any system
The laws are very general and make almost no assumptions about the behaviour of the random variables characterising the system
Another advantage of the laws is their simplicity: they can be applied quickly and easily
They don't follow common distributions and can be analysed analytically, they are very simple and can be applied very easily
Requests are not distinguishable so I study the mean responses



Operational laws are based on observable variables - values which we could derive from watching a system over a finite period of time

- We assume that the system receives requests from its environment
- Each request generates a job or customer within the system
- When the job has been processed the system responds to the environment with the completion of the corresponding request

  If we observed such an abstract system we might measure the following quantities:
- T, the length of time we observe the system
- A, the number of request arrivals we observe
- C, the number of request completions we observe
- B, the total amount of time during which the system is busy
- N, the average number of jobs in the system

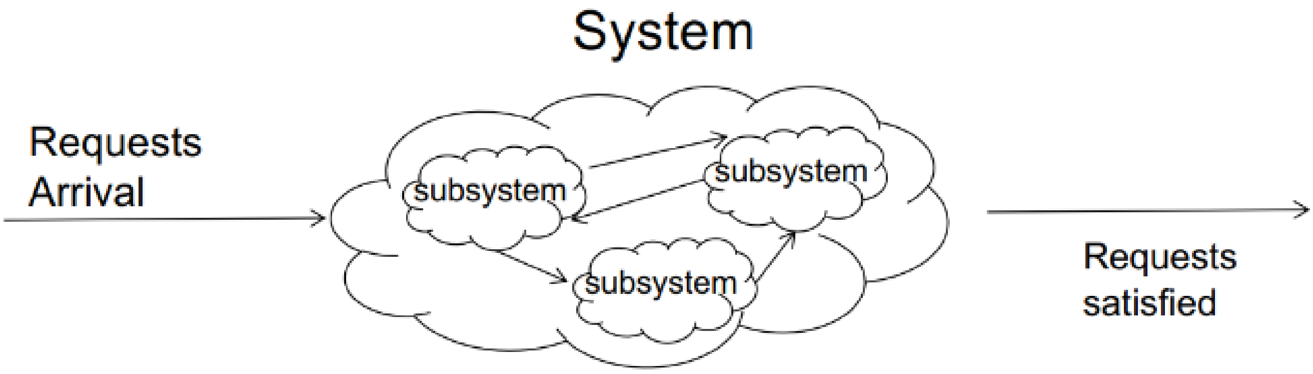
  From these observed values we can derive the following four important quantities:
- $\lambda = A/T$, the arrival rate
- $X = C/T$, the throughput or completion rate
- $U = B/T$, the utilisation(range is 0 to 1)
- $S = B/C$, the mean service time per completed job(time containing the waiting, the work time and the departure time)

  Single access perspective

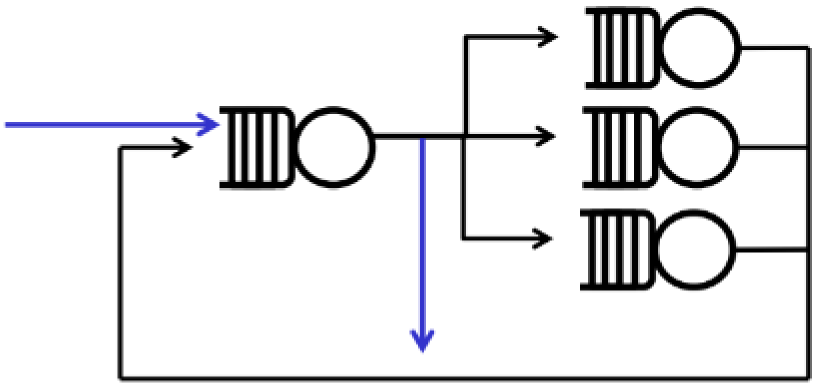
Job flow balance

We will assume that the system is job flow balanced

- This means that the number of arrivals is equal to the number of completions during an observation period, i.e., A = C
  This is a testable assumption because an analyst can always test whether the assumption holds
- It can be strictly satisfied by careful choice of measurement interval
  Note that if the system is job flow balanced the arrival rate will be the same as the completion rate, that is: $\lambda = X$



A system may be regarded as being made up of a number of devices or resources
Each of these may be treated as a system in its own right from the perspective of operational laws
An external request generates a job within the system; this job may then circulate between the resources until all necessary processing has been done; as it arrives at each resource it is treated as a request, generating a job internal to that resource



A system may be regarded as being made up of a number of devices or resources
Each of these may be treated as a system in its own right from the perspective of operational laws
An external request generates a job within the system; this job may then circulate between the resources until all necessary processing has been done; as it arrives at each resource it is treated as a request, generating a job internal to that resource

- T, the length of time we observe the system

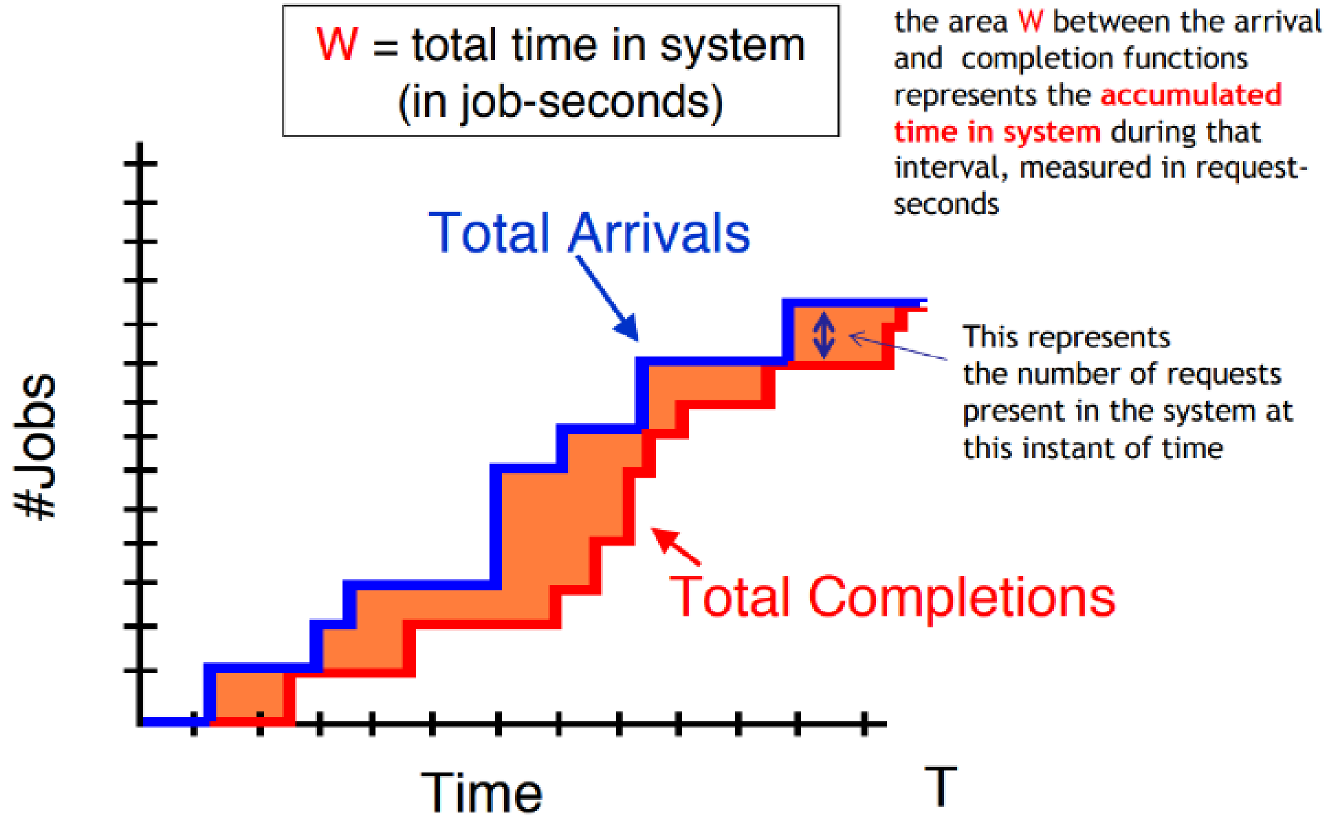
- A_k, the number of request arrivals we observe for resource k
- C_k, the number of request completions we observe at resource k
- B_k, the total amount of time during which the resource k is busy ($B_k \leq T$)
- N_k, the average number of jobs in the resource k (queueing or being served)
  From these observed values we can derive the following four important quantities for resource k:
- λ_k = A_k/T , the arrival rate
- X_k = C_k /T , the throughput or completion rate
- U_k = B_k/T, the utilisation
- S_k = B_k/C_k, the mean service time per completed job

Utilization law

Let us recall the following definitions for a resource k:
Throughput: $X_k = C_k/T$
Service time $S_k = B_k/C_k$
Utilization: $U_k = B_k/T$
From:
$X_kS_k = C_k/T * B_k/C_k = B_k/T = U_k$
we can derive (utilization law): $U_k = X_kS_k$

Little Law

Little’s law: N = XR
Little law can be applied to the entire system as well as to some subsystems
N = average number of requests in the system
If the system throughput is X requests/sec, and each request remains in the system on average for R seconds, then for each unit of time, we can observe on average XR requests in the system

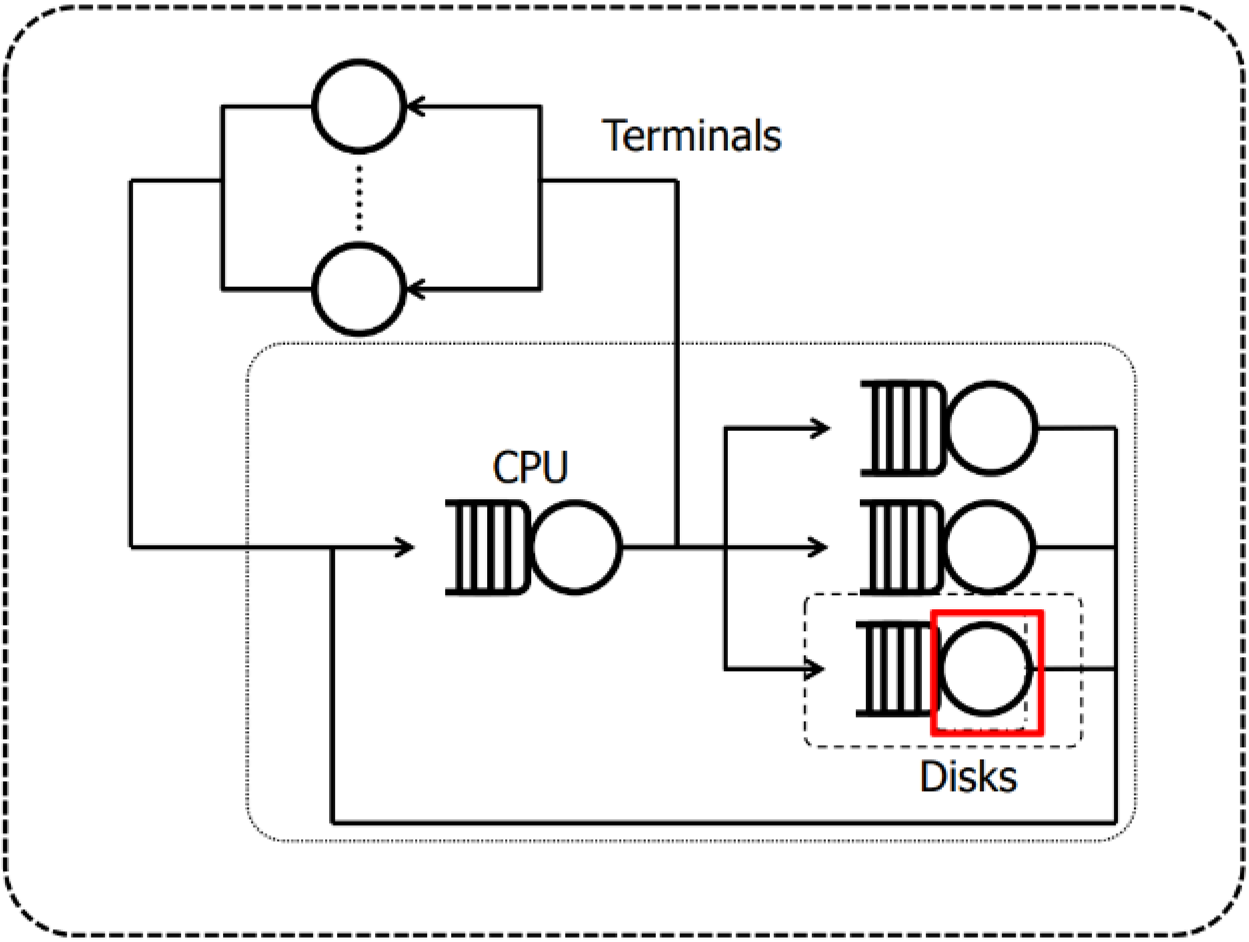

W = total time in system (in job-seconds)
the area W between the arrival and completion functions represents the accumulated time in system during that interval, measured in request-seconds
Total Arrivals
This represents the number of requests present in the system at this instant of time
Total Completions
#Jobs
Time
T

W denotes the accumulated time in system (jobs- sec)

- if there are 3 requests in the system during a 2 second period, then W is 6 request-seconds
  N the average number of requests in the system is: N=W/T
  R the average system residence time per request is: R=W/C
  We can write:
  N = W/T = C/T *W/C* = *X*R, so N = XR

Application of Little’s Law at different levels

Little’s Law, level 1



It can be applied to the single server Disk (without the queue) (Red Box)

$N_{(1)}$ in this case represents the percentage of time in which the server Disk is busy, so it corresponds to U_{disk}

$R_{(1)}$ represents the requests average service time

$X_{(1)}$ represents the rate of serving requests

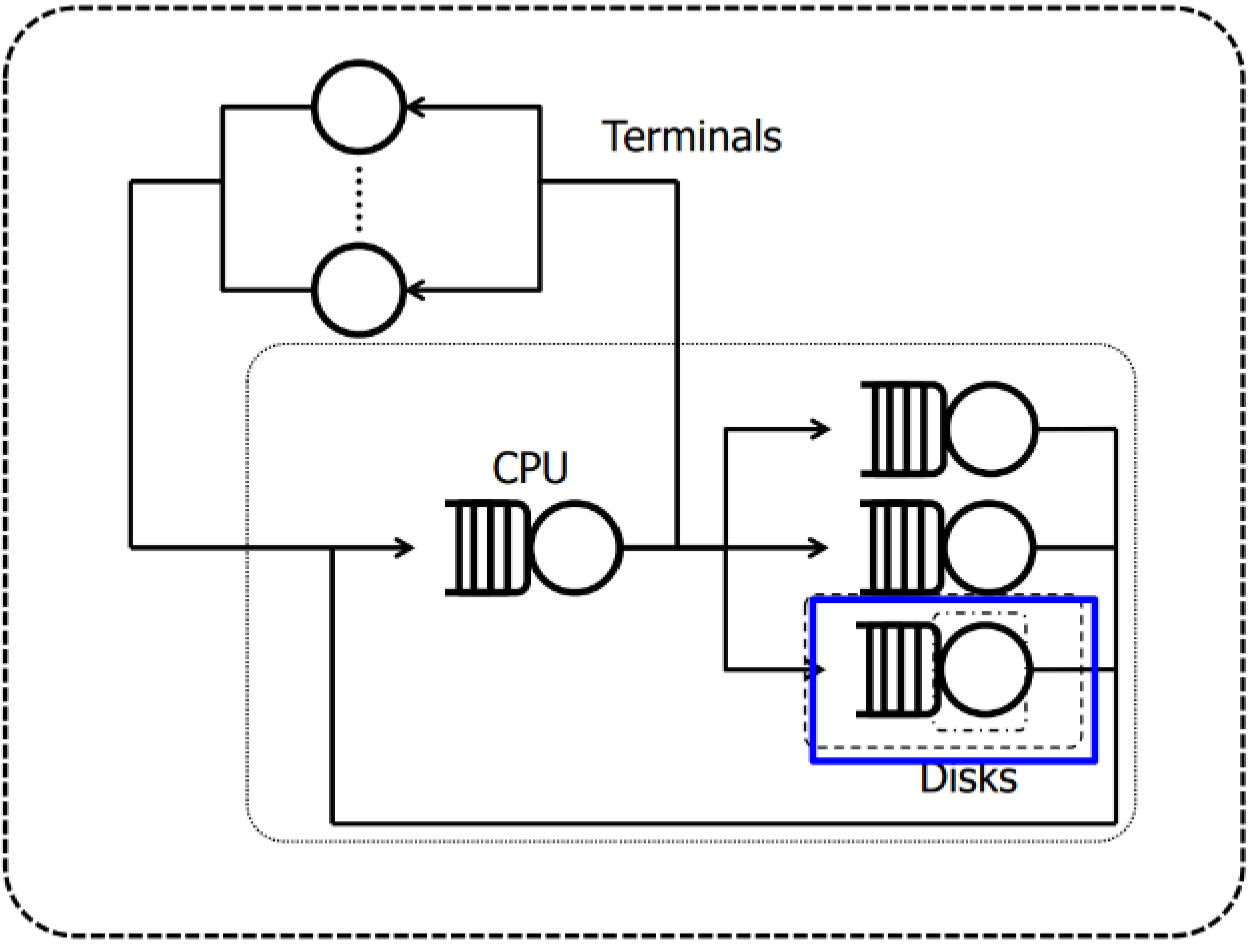
Let us apply Little law:

$X_{(1)}$ =40 req/sec, S =$R_{(1)}$ =0.0225 sec,

Little law $N_{(1)}$ =$X_{(1)}$ $R_{(1)}$ =0.9

$U_{(1)}$ =$X_{(1)}$ S, $U_{(1)}$ = 90%

Little’s Law, level 2



Let us include now the queue (Blue Box)

$N_{(2)}$ is the number of users in the service center (waiting + in service)

$R_{(2)}$ is the time spent in the service center (waiting time + service time)

$X_{(2)}$ is the throughput of the server Disk and corresponds to $X_{(1)}$

We know that $X_{(1)}$=$X_{(2)}$ = 40 req/s.

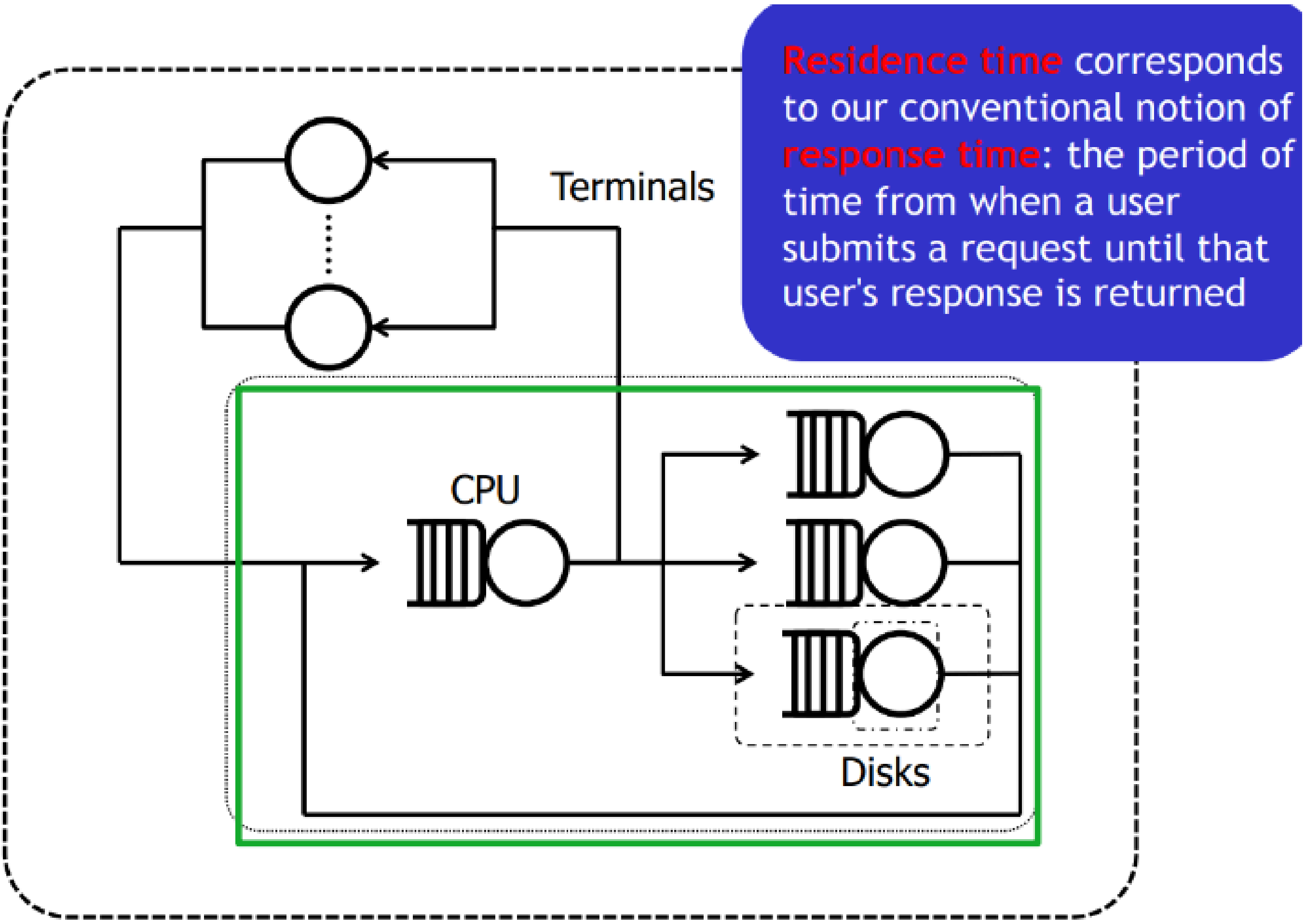
Let us suppose to have obtained, through observations, the following information: $N_{(2)}$ = 4.

From Little's law we have: $R_{(2)}$ =$N_{(2)}$/$X_{(2)}$= 0.1 s.

Since $R_{(1)}$ = 0.0225 s, we can compute the waiting time as:

$R_{(2)} - R_{(1)}$ = 0.0775 s

Little's Law, level 3



Residence time corresponds to our conventional notion of response time: the period of time from when a user submits a request until that user's response is returned

Let us consider the central subsystem (Green Box)

$N_{(3)}$ represents the total number of users in the subsystem (e.g., requests of web pages/s)

$R_{(3)}$ represents the average time spent in the subsystem by each request

$X_{(3)}$ is the subsystem throughput (e.g., number of web pages/s)

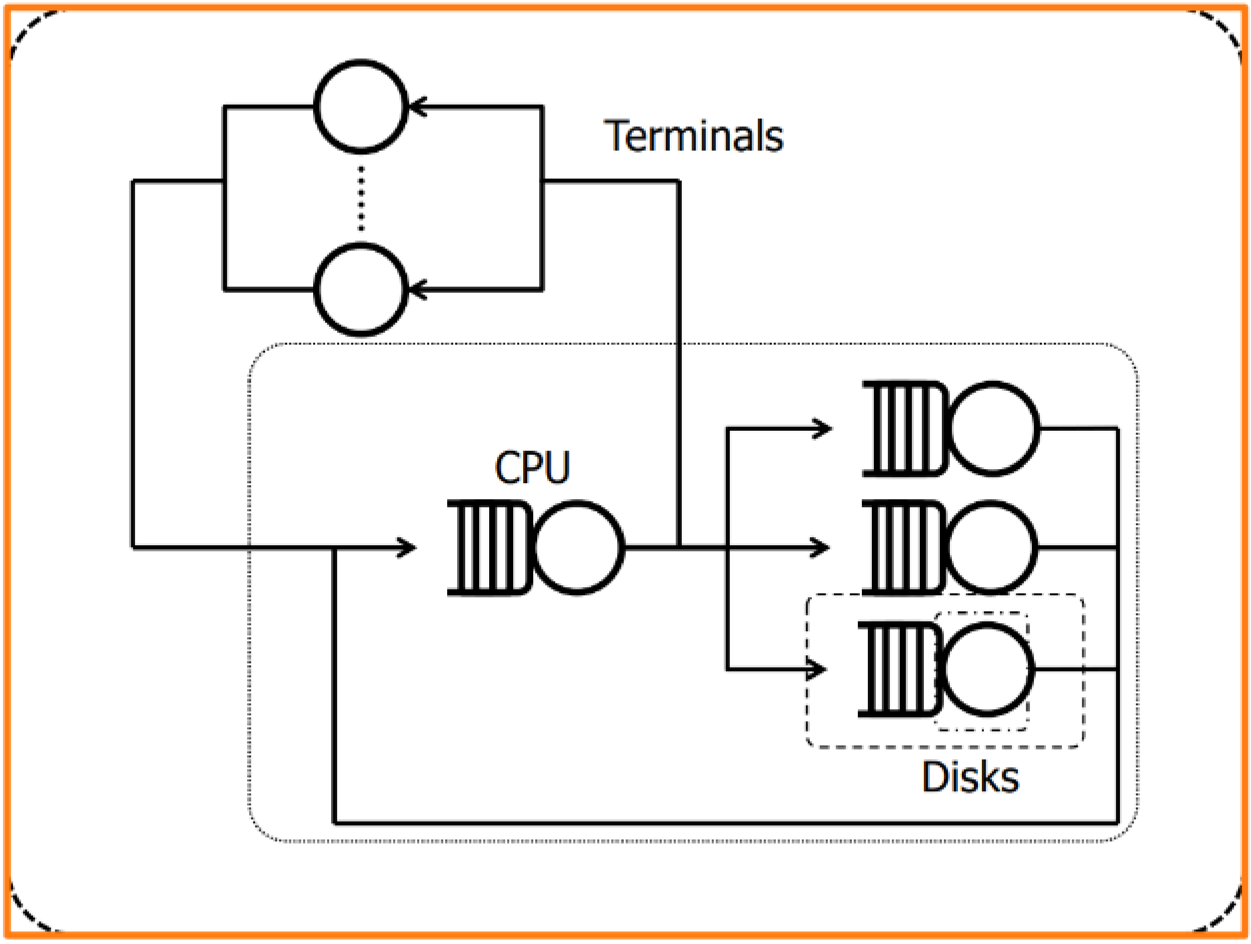
If from observations we know that $X_{(3)}$ = 0.5 job/sec and

requests number is $N_{(3)}$ = 7.5

From Little law we have that

$R_{(3)} = N_{(3)}/X_{(3)} = 15s$

Little’s Law, level 4

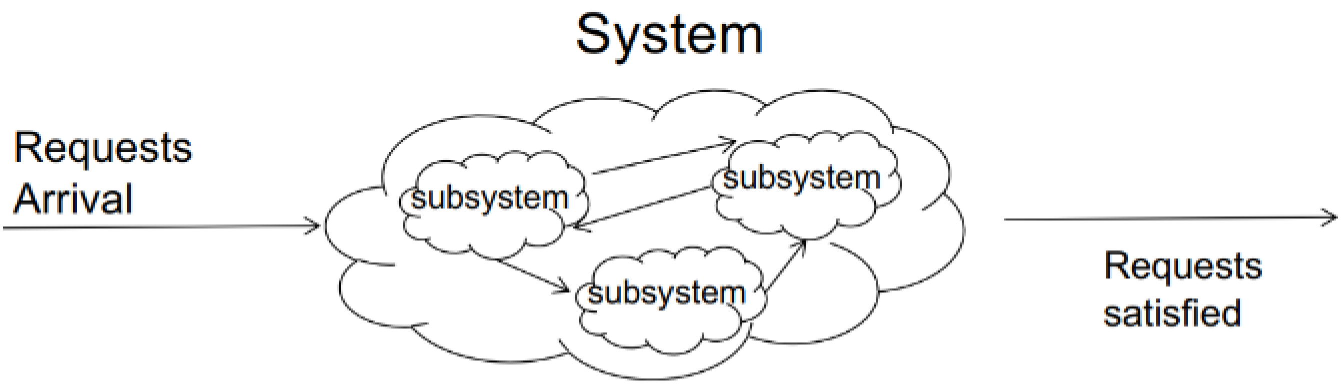


Let us now consider the complete system (Orange Box)
N(4) is the total number of users in the system (which is fixed since we have a closed system)
R(4) is the total amount of time spent in service, waiting and at the "terminals" client side (think time, e.g., the time a user spend to read a web page and to elaborate a request)
X(4) is the rate with which the requests reach the systems from the terminals client and it corresponds to X(3)
Let us suppose that there are N(4) = 10 users, and the think time is Z = 5 s.
We know that the time spent in the system is R(3) = 15 s.
Now, since R(4) = R(3) + Z, we can derive from Little law that
N(4) = X(4)(R(3) + Z),
and we can compute
X(4) = 0.5 job/s

Interactive Response Time Law

Back when most processing was done on shared mainframes, think time, Z, was quite literally the length of time that a programmer spent thinking before submitting another job
More generally in interactive systems, jobs spend time in the system not engaged in processing, or waiting for processing: this may be because of interaction with a human user, or may be for some other reason
The think time represents the time between processing being completed and the job becoming

available as a request again
Interactive Response Time Law: R = N/X − Z
The response time in an interactive system is the residence time minus the think time
Note that if the think time is zero, Z = 0 and R = N/X , then the interactive response time law simply becomes Little’s Law
Suppose that the library catalogue system has 64 interactive users connected via Browsers, that the average think time is 30 seconds, and that system throughput is 2 interactions/second. What is the response time? The interactive response time law tells us that the response time must be 64/2 − 30 = 2 seconds



In an observation interval we can count not only completions external to the system, but also the number of completions at each resource within the system
Let C_k be the number of completions at resource k
We define the visit count, V_k , of the k-th resource to be the ratio of the number of completions at that resource to the number of system completions $V_k = C_k/C$
Visits
Note that:

- If $C_k > C$, resource k is visited several times (on average) during each system level request. This happens when there are loops in the model
- If $C_k < C$, resource k might not be visited during each system level request. This can happen if there are alternatives (i.e. caching of disks)
- If $C_k = C$, resource k is visited (on average) exactly once every request

Forced Flow Law

The forced flow law captures the relationship between the different components within a system. It states that the throughputs or flows, in all parts of a system must be proportional to one another.

$$X_k = V_k X$$

The throughput at the k-th resource is equal to the product of the throughput of the system and the visit count at that resource.
Rewriting $C_k = V_k C$ and applying $X_k = C_k/T$, we can derive the forced flow law:

- $C_k = V_k C$
- $C_k/T = V_k C/T$
- $X_k = V_k X$

Utilisation Law

If we know the amount of processing each job requires at a resource then we can calculate the utilisation of the resource
Let us assume that each time a job visits the k-th resource the amount of processing, or service time it requires is S_k
Note that service time is not necessarily the same as the response time of the job at that resource: in general a job might have to wait for some time before processing begin
The total amount of service that a system job generates at the k-th resource is called the service demand, D_k: $D_k = S_k V_k$
The utilisation of a resource, the percentage of time that the k-th resource is in use processing to a job, is denoted U_k.
Utilisation Law: $U_k = X_k S_k = (XV_k)S_k = D_k X$
The utilisation of a resource is equal to the product of:

1. the throughput of that resource and the average service time at that resource
2. the throughput at system level and the average service demand at that resource
   $U_k = D_k X$
   Note that:

- Average service time Sk accounts for the average time that a job spends in station k when IT IS SERVED
- Average service demand Dk accounts for the average time a job spends in station k during ITS STAYING IN THE SYSTEM. As seen for the visits, depending on the way in which the jobs move in the system, the demand can be less than, greater than or equal to the average service time of station k

Response and Residence times

- Response time: time spent at a resource in a single time($\tilde{R}_k$)
- Residence time: total time spent at a single resource(R_k)
  Note that there is the same relation between Residence Time and Response Time as the one between Demand and Service Time

  $$D_k = v_k S_k, R_k = v_k \tilde{R}_k$$

  Also note that for single queue open system, or tandem models, $v_k = 1$.
  This implies that average service time and service demand are
  equal, and response time and residence time are identical:

  $$v_k = 1 \Rightarrow D_k = S_k, R_k = \tilde{R}_k$$

General Response Time Law

One method of computing the mean response time per job in a system is to apply Little's Law to the system as a whole
However, if the mean number of jobs in the system, N, or the system level throughput, X, are not known an alternative method can be used
Applying Little's Law to the k-th resource we see that $N_k = X_k \tilde{R}_k$, where N_k is the mean number of jobs at the resource and $\tilde{R}_k$ is the average time spent at the resource (for the single interaction at the k-th resource level, e.g. disk request)
From the Forced Flow Law we know that $X_k = XV_k$. Thus we can deduce that:
$N_k/X = V_k \tilde{R}_k = R_k$
The total number of jobs in the system is clearly the sum of the number of jobs at each resource, i.e. $N = N_1 + \ldots + N_M$ if there are M resources.
From Little's Law R = N/X and so, from $R_k = N_k/X = V_k \tilde{R}_k$
General Response Time Law: $R = \sum_k V_k \tilde{R}_k = \sum_k R_k, R_k = V_k \tilde{R}_k, \forall k$
The average response time of a job in the system is the sum of the product of the average time for the individual access at each resource and the number of visits it makes to that resource
The average response time of a job in the system is the sum of the resources residence time

Model Evaluation

Using Little's Law we calculate the time spent at each disks (remembering that the number in the system is the number in the buffer +1): $\tilde{R}_{diskA}$, $\tilde{R}_{diskB}$
Then: $R = \tilde{R}_{CPU} V_{CPU} + \tilde{R}_{diskA} V_{diskA} + \tilde{R}_{diskB} V_{diskB}$
Using Little's Law we calculate the time spent at each disks (remembering that the number in the system is the number in the buffer +1): $\tilde{R}_{diskA} = N_{diskA}/X_{diskA}$, $\tilde{R}_{diskB} = N_{diskB}/X_{diskB}$
Then: $R = \tilde{R}_{CPU} V_{CPU} + \tilde{R}_{diskA} V_{diskA} + \tilde{R}_{diskB} V_{diskB}$

Operational laws

Operational laws are simple equations which may be used as an abstract representation or model of the average behaviour of almost any system
The laws are very general and make almost no assumptions about the behaviour of the random variables characterising the system
Another advantage of the laws is their simplicity: this means that they can be applied quickly and easily