**GOYA AMEBE-OBARI AWIANKELE**

**16/SCI03/007**

**CSC 410**

1) Concept of Operational Lawa in Computer Systems -

Operational laws are equations which may be used as an abstract representation of a model of the average behavior of a system. They are general and make no assuptions about the behavior of a random variable that characterize the system. Using these abstactions, these laws can be applied to any device and can slowly build a more complex system.

2) Operational laws employed in computer system performance evaluation;

a) **Interactive Response Time Law**; the name of this law goes back to the time when most of the systems which were being modelled were mainframes processing both interactive jobs and batch jobs. The think time 'Z', was quite the length of time that a programmer spent thinking at his terminal before summiting another job. Interactive systems are those in which jobs spend time in the system not engaged in processing or waiting for processing: this maybe because of interaction with a human user, or maybe because of some other reason.

The think time is the time between processing being completed and the job becoming available as a request again. The residence time of the job is the time from arrival to completion; this is greater the users response time. The interactive response time reflects this: it callculates the response time, R as follows: 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 i zero, Z=0 and R=W, the the response time law simply becomes Little's law.

b) **Little's Law**; this is a theorem that determines the average number of items in a stationary queuing system based on the average waiting time of an item within a system and the average number of items arriving at the system per unit time.

This law provides a simple and intuitive approach for the assessment of the efficiency of queuing systems. The concept is hugely significant for business operations because it states that the number of items in the queuing system primarily depens on two key variables, and it is not affected by other factors such as the distribution of the service order.

Little can only be used in queuing systems. Formula for Little's law; L=sigma \* w

where L is the average number of items in a queuing system

Sigma is the average number of items arriving at the system per unit of time

W is the average waiting time an item spends in a queuing system.

c) **Conservation Law**; Whatever goes in must come out. If a system with the arrival rate of ***pie*** and an output rate of ^. If the system is not overloaded and customers are neither created nor destroyed inside the system, then ***pie*** = ^ . The process of creating customers in a system is called forking and the process of destroying customers is called joining. It is possible to model systems with these behaviors, but usually difficult.

d) **General Residence Time Law**; One of the methods of computing the mean residence or 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 alterative method can be used. Applying Little's law to the ith resource we see that Ni = XiWi, where Ni is the mean number of jobs at the resource and Wi is the average response time of the resource. From the forced flow law we know that Xi = XVi. Hence we can say that Ni/X = ViWi.

The total number of jobs in the system is clearly the sum of the number of jobs at each resource, i.e, N = N1 + . . . + NM if there are M resources in the system. We know from Little's law that W = N/X. The average residence time of a job in the system will be the sum of the product of its average residence time at each resource and the number of visits it makes to that resource.

e) **Space-Time Product Law**; This states that the throughput is equal to average amount of memory in use divided by average space-time product. Space Time Products are often used to evaluate program performance and assign accounting charges to programs in virtual memory systems. Essentially, a program's space-time product is equal to its execution time multiplied by the average amount of money allocated to it during its execution. Since space-time products, response time, and throughput are all used as indicators of system performance, it is interesting to examine the manner in which these quantities are related.

f) **Utilization**; This states that the utilization of a resource is equal to the product of the throughput of that resource and the average service requirement at that resource. If we know the amount of processing that each job requires at a resource then the utilization of the resource can be calculated. The total amount of service that a system job generated at the ith resource is called the service demand, Di: Di = SiVi

The utilization of a resource, the percentage of time that the ith resource is in use processing to a job, is donated Ui

 Ui = XiSi = XD

g) **Forced Flow Law**; This relates throughputs at individual resources within a system to the overall system throughput. It is the average number of visits that a system level job makes to that resource. The general residence time law is the sum of the product of its average residence time at each resource and the number of visits it makes to that resource.

4. Some basic queuing models and basic queuing disciplines include;

i) **First-in-first out (FIFO)**; the oldest invetory items are recorded as sold first but do not necessarily mean that the exact oldest physical object has to be tracked and sold i.e, the cost associated with the inventory that was purchased first is the cost expensed first.

ii) **Last-in-first-out (LIFO)**; this simply describes a method for accounting for inventories. In this system, the last unit added to an inventory is the first to be recorded as sold.

iii) **Service-in-random-order (SIRO)**; in this queue structure, the customer is chosen randomly and hence all customers are equally likely to be selected. Therefore, the time of arrival of the customer has no consequence on the selection of the customer.

iv) **Shortest-processing-time-first (SPT)**; its principle is to order jobs according to their duration and schedule them by beginning by the shorters.