CS 547 Lecture 3: Little's Result

Daniel Myers

Little's Result (sometimes called Little's Law) is the key equation for understanding system dynamics. We'll use it over and over again to derive new results in a variety of contexts.

The equation relates three key system parameters: throughput, residence time, and the average number in the system.

$$\overline{N} = \Lambda \overline{R}$$

For example, if a university awards 50 CS bachelor's degrees per year, and it takes the average student four years to earn the degree, we would expect an average of 200 undergraduate students in the CS program.

Proof

As with the Forced-Flow Law, we'll make a measurement-based argument for the validity of Little's Result.

Suppose we measure the system for a time period of length T and record C completions during the interval. Now, divide the entire measurement interval into non-overlapping periods where the number of customers in the system is stable. When an arrival or departure occurs, the number of customers in the system changes, then remains stable until the next arrival or departure occurs.

If the i^{th} stable period lasts for t_i and has a population of n_i customers, we can define the total time accumulated by all customers in the during the period (measured in units of customer-seconds) as

$$\gamma_i = n_i t_i.$$

The total number of customer-seconds accumulated over the entire measurement interval is

$$\gamma = \sum_i \gamma_i$$

Now, the following relationships hold:

$$\overline{N} = \frac{\gamma}{T}$$
$$\overline{R} = \frac{\gamma}{C}$$
$$\Lambda = \frac{C}{T}$$

Using the relationships, we can verify that $\overline{N} = \Lambda \overline{R}$ to complete the argument.

The Power of Little's Result

Little's Result is an extremely useful result for two reasons.

First, it's an extremely *general* result. It makes no assumptions about how arrivals occur, how customers accumulate service, or any other statistical properties of the system. Little's Result applies to *any* system.

Second, we can apply the equation to any *part* of a system. As we go through this course, we'll use Little's Result to analyze entire systems of queues, individual queues, groups of queues within a system, and even parts of a single queue. The same relationship between throughput, occupancy, and residence time holds in all parts of a system.

Examples

We measured a router for 30 minutes. 3.6×10^6 *packets were transmitted during the interval and the average queueu length was 3 packets, including the packet in service. Find the average residence time.*
This is a straightforward application of the equation: given two of the values, find the third.

$$
\begin{aligned}
\overline{R} &= \frac{\overline{N}}{\Lambda} \\
&= \frac{3\, packets}{3.6 \times 10^6\, packets \,/\, (30 * 60)\, sec} \\
&= .0015\, s
\end{aligned}
$$

A job trace records the arrival time, start time, and completion time for each task in a system. What is the average number of waiting jobs?
We can find the average number waiting by applying Little's Result to the queue's waiting line. The average time spent in line is simply the average waiting time, $\overline{W}$.

$$\overline{N}_{waiting} = \Lambda \overline{W}$$

The average waiting time can be calculated by comparing the arrival time of each job to its start time. The thoughput can be calculated from the arrival times.

The Utilization Law

We can also apply Little's Result to the server of the queue. The average time spent at the server is simply the average service time, $\overline{s}$.

$$\overline{N}_{server} = \Lambda \overline{s}$$

The average number in the server is equivalent to the fraction of time the server is busy serving a customer. This is the utilization, U. Therefore, the Utilization Law is simply a special case of Little's Result:

$$U = \Lambda \overline{s}.$$

Under normal operation, $0 \leq U \leq 1$. If the system is overloaded (when $\lambda > \mu$), the calculation will result in $U > 1$.

Corollary: If $\overline{Q}$ is the average number in the queue including the customer in service, $\overline{Q} - U$ is the average number waiting and not being served.

Combining the Forced-Flow Law and Little's Result

Consider a desktop computer with one disk that serves page faults. We measured that $U_{disk} = .8$, $\overline{s}_{disk} = .01$ *sec, and* $\Lambda_{system} = 5$ *tasks per second. What is the average number of page faults per task?*

We need to calculate $\overline{V}_{disk}$. First use the Utilization Law to find the disk throughput.

$$\begin{aligned}\Lambda_{disk} &= \frac{U}{\overline{s}} \\ &= \frac{.80}{.01} \\ &= 80\,\frac{faults}{sec}\end{aligned}$$

Now, use the Forced-Flow Law to find $\overline{V}_{disk}$.

$$\begin{aligned}\overline{V}_{disk} &= \frac{\Lambda}{\Lambda_{disk}} \\ &= \frac{80}{5} \\ &= 16\,\frac{faults}{task}\end{aligned}$$

Multiple Classes

In some applications, we divide the customers in a system into multiple classes. For example, we might use two classes to model different levels of I/O service, where one class gets priority treatment and the other only gets best-effort service.

If we have two classes, A and B, with paramters $\overline{N}_A$, Λ_A, $\overline{N}_B$, Λ_B, then the numbers of customers and throuhgputs are additive.

$$\begin{aligned}\overline{N} &= \overline{N}_A + \overline{N}_B \\ \Lambda &= \Lambda_A + \Lambda_B\end{aligned}$$

To calculate the average residence time, we can use Little's Result to weight the class residence times.

$$\begin{aligned}\Lambda\overline{R} &= \overline{N} \\ &= \overline{N}_A + \overline{N}_B \\ &= \Lambda_A\overline{R}_A + \Lambda_B\overline{R}_B\end{aligned}$$

$$\overline{R} = \frac{\Lambda_A}{\Lambda}\overline{R}_A + \frac{\Lambda_B}{\Lambda}\overline{R}_B$$

Closed Systems

Little's Result still applies to closed systems, but we have to be more careful about defining what "residence time" means in a system where customers never really leave.

Suppose we pick one particular point in the system that's visited by all customers, then consider the total time required for a customer toleave the point, travel all the way through the closed loop, and return to the point. There are two things the customer must do on its trip through the loop: visit the think node for an average time of $\overline{Z}$ and visit the rest of the system for an average time of $\overline{R}$. Thus, the average time a customer spends on one trip through the loop is $\overline{R} + \overline{Z}$.

Since the number of customers N is a fixed parameter, we can use Little's Result to calculate throughput:

$$\Lambda = \frac{N}{\overline{R} + \overline{Z}}.$$

This result matches our previous definition of throughput in a closed system.