CS 547 Lecture 4: Bounds on Performance

Daniel Myers

In this lecture, we used the fundamental laws to derive several basic bounds on performance, using only simple parameters.

Demand

Define the *demand* at resource k to be the total average service time accumulated at the resource. That is,

$$\overline{D}_k = \overline{V}_k \overline{s}_k.$$

In some cases, it can be easier to measure average demand than it is to measure visit counts and service times.

Bottlenecks

We can combine the Utilization and Forced-Flow Laws to obtain a bound on system throughput.

$$\begin{aligned} U &= \Lambda_k \overline{s}_k \\ &= \overline{V}_k \Lambda \overline{s}_k \\ &= \Lambda \overline{D}_k \end{aligned}$$

Under normal operation $U \leq 1$, so we can bound the system throughput in terms of demand:

$$\Lambda \leq \frac{1}{\overline{D}_k}.$$

This bound is tightest at the resource with maximum demand, so

$$\Lambda \leq \frac{1}{\overline{D}_{max}}.$$

Consider a system with 100 memory modules. Each job generates one request to each memory module, plus one additional request to module 1. Bound the system throughput.
The first module is clearly the bottleneck, and we have $\overline{V}_1 = 2$. The bound is

$$\Lambda \leq \frac{1}{2\overline{s}_{mem}}.$$

If we balanced the loads so that the one extra request was equally distributed across all 100 modules, we would have $\overline{V} = 1.01$ for all modules and a throughput bound of

$$\Lambda \leq \frac{1}{\overline{s}_{mem}}.$$

Balancing the load nearly doubles throughput!

The bottleneck bound gives rise to the most obvious system design tip of all time: *alleviate the bottleneck!*

Corollary: in a balanced system, improving an individual resource will not have a significant effect on performance, because the unimproved resources will still constrain throughput.

Closed System Bounds

The bottleneck bound applies to all systems, but we can derive other useful bounds in closed systems.

Recall the definition of throughput in a closed system,

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

To find an upper bound on Λ, we should find the minimum feasible value for $\overline{R}$.

The fastest a customer can make it through the service system is to visit every center and receive service without experiencing any queueing delay. The expected time for such a trip is simply the sum of demands at all the service centers.

$$\overline{D}_{total} = \sum_k \overline{V}_k \overline{s}_k$$

This value is a lower bound on the residence time, so we can use it bound the throughput as a function of N,

$$\Lambda \leq \frac{N}{\overline{D}_{total} + \overline{Z}}.$$

The bottleneck bound still applies to the entire system, so the actual maximum throughput is

$$\Lambda \leq \min\left(\frac{N}{\overline{D}_{total} + \overline{Z}},\ \frac{1}{\overline{D}_{max}}\right).$$

Which of the two bounds holds depends on the value of N. A good feasible operating point is at the intersection of the two bounds. Setting the two bounds equal and solving for N yields

$$N^* = \frac{\overline{D}_{total} + \overline{Z}}{\overline{D}_{max}}.$$

The $\overline{D}_{total}$ bound applies when $N < N^*$. When $N > N^*$, the bottleneck bound holds.

Bounding the Residence Time

Applying Little's Result to a closed system gives an equation for $\overline{R}$ in terms of N, Λ, and $\overline{Z}$,

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

Making Λ as large as possible yields a lower bound on residence time. The maximum value of Λ is given by the bound in the previous section.

$$\overline{R} \geq \frac{N}{\Lambda_{max}} - \overline{Z}.$$