THROUGHPUT

Peter J. Denning, Naval Postgraduate School, Monterey, California

January 2008
Rev 5/26/08

Abstract: The throughput of a system is the number of jobs, tasks, or transitions they system completes per unit time. It is often used as a figure of merit, with higher values indicating better performance.

Keywords: throughput, response time, congestion, bottlenecks, queueing networks

The throughput of a system is the number of jobs, tasks, or transactions the system completes per unit time. If during an observation period of length T the system completes C jobs, throughput X is measured as C/T. Throughput is often used as a figure of merit, with higher values indicating better performance. System engineers use analytic or simulation models to help determine a system configuration and server capacities that enable the system to meet throughput and response-time targets.

The term "throughput" suggests a stream of jobs flowing *through* a system. A through job would arrive and depart during the observation period. Completing jobs that were already in the system at the start of the observation period would not be counted. Therefore, counting all completions in the observation period overestimates throughput. However, in most systems, there is an upper limit N on the number of jobs in the system, reflecting the system capacity or the size of the user population. Therefore, the number of completions C is within N of the number of through jobs. For a long observation period, C is large compared with N, and throughput measured as X = C/T is negligibly different from "true" throughput.

Throughput considered alone is often a deceptive measure of performance. One common deception is a server with a long queue: Its throughput is at saturation. Although the saturation throughput may be acceptable, the server's response time would be unacceptably high. Acceptable throughput does not guarantee acceptable response time.

Another deception can occur in a multiserver system. At a high load, the system's bottleneck saturates and produces its maximum possible throughput. In turn that limits the throughput at every other server in the system. If a nonbottleneck is used as the throughput reference point, the unacceptable

throughput there will be due to the bottleneck, not to that server. Increasing the speed of the nonbottleneck server will not increase saturation throughput.

The subtle relationships among throughput, response time, and bottlenecks can be understood with the help of four fundamental laws of networked systems (1). The parameters are as follows:

Vi = number of visits per job to server i
Si = means service time per visit at server i
N = number of users in the system
Z = think time of a user before making a new request of the system

The laws are:

- *Utilization Law*: A server's utilization (fraction of time busy) is the product of its service time and throughput: $U_i = S_i \times X_i$.
- *Little's Law*: a server's mean queue length is the product of its response time per visit and throughput: $Q_i = R_i \times X_i$.
- *Forced Flow Law*: A server's throughput is the product of the system throughput and visit ratio: $X_i = X \times V_i$.
- *Response Time Law*: System response time per job plus think time is the number of jobs divided by the system throughput: $R + Z = N/X$.

A system bottleneck analysis follows immediately from these laws.

- Define demand $D_i = V_i \times S_i$ as the total expected job service required over all visits to a server.
- Since utilization cannot exceed 1, the utilization law implies $X_i \leq V_i/D_i$.
- Combining with the forced flow law, the system throughput $X \leq 1/D_i$ for all servers.
- Therefore, the server with largest D_i limits the system throughput and is the bottleneck.
- Combining with the response time law, the mean response time cannot be smaller than $D_i \times N - Z$.

An example will help illustrate these concepts. Consider a two-server system in which a job's total CPU demand is 2 seconds, a typical job visits a DISK server 100 times for 50 milliseconds each, and there are 10 users with an average think time of 20 seconds. The CPU demand is $D1 = 2$ seconds and the DISK demand is $D2 = 5$ seconds. The DISK is the bottleneck, and system throughput is limited to $1/5$ job per second. The CPU utilization cannot be higher than $D1/D2 = 2/5$. A faster CPU will have a smaller D1 -- speeding it up will only *reduce* its utilization. Speeding up the DISK will increase system throughput and CPU utilization. The system response time is at least $5 \times 10 - 20 = 30$ seconds. A response time of less than 30 seconds for 10 users is impossible.

The bottleneck analysis sketched above is for systems in which the parameters Vi and Si are independent of the load N in the system. If one Vi depends on N, then

the upper bound on throughput may depend on N. An example of this occurs in multiprogrammed virtual memories: As load N increases, job memory allocations are squeezed and paging traffic (Vi for the paging disk) increases. As load N increases, the total demand for the paging disk eventually exceeds all others and the paging disk becomes the bottleneck, forcing throughput down from a peak. (See the "thrashing" article.)

These simple relationships can be extended to systems with multiple job classes and variable rate servers (2).

BIBLIOGRAPHY

1. P. Denning and J. Buzen, Operational Analysis of queueing network models, ACM *Comput. Surv.* **25** (September): 225-261, 1978.

2. D. Menascé, et al., *Capacity Planning*. Englewood Cliffs, NJ: Prentice-Hall, 1994.