Transaction Performance

COMP755 Advanced OS

Goals

- Estimate the performance or capacity of a computing system
- Understand how to effectively use the information provided by a system monitor

Transaction Systems

- A transaction is typically a predefined program that is executed when an input arrives. The transaction usually performs some processing and then sends output back to the user.
- Transactions usually terminate or wait for more input after performing their short function.

Transaction Examples

- Web servers
  - Static web pages
  - Search engines
- Web services
- Credit card authorization

- We are NOT concerned, at this time, with the performance of a laptop or home system.

Measuring Performance

- Most operating systems provide tools to assist in measuring system performance
- Programs can be written to take simple measurements of activity and performance

Easily Measurable Values

- Length of an observation period.
- Number of jobs during the observation period.
- Number of accesses for each device
- % utilization of the CPU
- Hardware service times

Job Flow

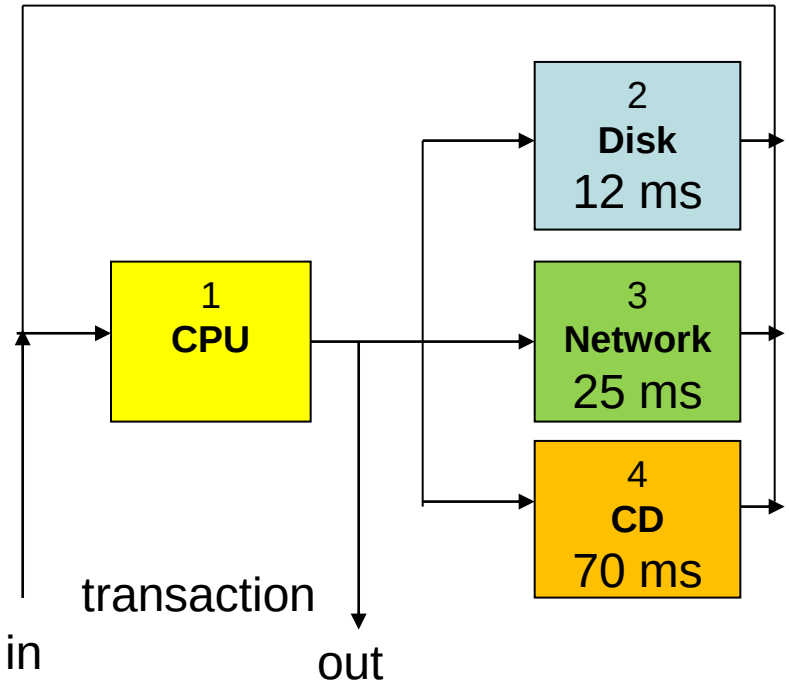
- When a program is run, it will generally spend some time using the CPU, then perform an I/O request
- When the I/O is done, the program will use the CPU for a while and then make another I/O request.
- This repeats (sometimes changing I/O devices) until the program uses the CPU and finishes

Execution Time

- The time required for a program to complete is the sum of the time at each device.
  - CPU time
  - I/O time
- The time can be calculated by multiplying the number of times a transaction visits each device (including CPU) by the average time spend at each device

$$TotalTime = \sum \#visits * avgDeviceTime$$

Example System



Transactions executed:	3,690
requests to the disk:	25,830
requests to the network:	22,140
requests to the CD:	7,380
CPU utilization:	32.8%

The time values represent the average service time for that unit. A performance monitor was run for a 15 minute period. The above data was collected.

Arrival Rates

- The arrival rate is the number of accesses to a device per time period. *(i.e. a disk might be used 34 times per second.)*
- Arrival rate for device i will be represented by the Greek letter lambda, λ_i
- Let λ_0 represent the arrival rate of jobs into the system *(i.e. web page requests)*
- Arrival rates can be calculated from the number of accesses divided by the length of the observation period.

Example Arrival Rates

The observation period is 15*60=900 seconds

Transactions executed:	3,690	4.1
requests to the disk:	25,830	28.7
requests to the network:	22,140	24.6
requests to the CD:	7,380	8.2
CPU utilization:	59,040	65.6

The CPU access count is the sum of the device and transaction counts.

Visitation Ratios

- The visitation ratio is the number of times an average program uses a device.
- The visitation ratio can be calculated by dividing the access rate of a device by the job entry rate. (i.e. disk/sec ***/*** webpage/sec)

$$V_i = \frac{\lambda_i}{\lambda_0}$$

Example Visitation Rates

Device	access	λ_i	V_i
Transactions executed:	3,690	4.1	
requests to the disk:	25,830	28.7	7
requests to the network:	22,140	24.6	6
requests to the CD:	7,380	8.2	2
CPU utilization:	59,040	65.6	16

Utilization

- The utilization of a device is the fraction of time the device is busy.
- You can compute individual devices utilization from the devices arrival rate and the mean service time.

$$\rho_i = \lambda_i * S_i$$

Note that utilization is always a number between zero and one.

Example Utilization

Device	access	λ_i	V_i	S_i	ρ_i
Transactions executed:	3,690	4.1			
requests to the disk:	25,830	28.7	7	0.012	.344
requests to the network:	22,140	24.6	6	0.025	.615
requests to the CD:	7,380	8.2	2	0.070	.574
CPU utilization:	59,040	65.6	16	0.005	**.328**

Saturation

- One device will typically saturate before the other devices when the load increases. This is the bottleneck device.
- A device saturates when its utilization is100%.
- From the service time of a device, you can calculate the arrival rate that will cause it to saturate.

$$\lambda_i = 1/S_i$$

- The busiest device saturates first.

Example Saturation

Device	access	λ_i	V_i	S_i	ρ_i
Transactions executed:	3,690	4.1			
requests to the disk:	25,830	28.7	7	0.012	.344
requests to the network:	22,140	24.6	6	**0.025**	**.615**
requests to the CD:	7,380	8.2	2	0.070	.574
CPU utilization:	59,040	65.6	16	0.005	.328

The network is busiest and will saturate with an arrival rate of 1/0.025 = 40.

Program Saturation

- We frequently want to know the number of jobs that can be run at saturation. This defines the maximum amount of work the system can perform.
- From the visitation ratio formula $\lambda_0 = \lambda_i / V_i$
- Using the saturation access rate $\lambda_i = 1/S_i$
- Combining them gives

$$\lambda_0 = \frac{1}{S_i * V_i}$$

Example Saturation

Device	access	λ_i	V_i	S_i	ρ_i
Transactions executed:	3,690	4.1			
requests to the disk:	25,830	28.7	7	0.012	.344
requests to the network:	22,140	24.6	**6**	**0.025**	**.615**
requests to the CD:	7,380	8.2	2	0.070	.574
CPU utilization:	59,040	65.6	16	0.005	.328

Based on network saturation, the maximum throughput is 1/(0.025*6) = 6.67

Minimum Execution Time

- If you know the service times but not the response time, you can calculate a minimum execution time. The average program may take longer, but will not complete sooner.

$$\min TransTime = \sum V_i * S_i$$

Example Transaction Time

Device	V_i	S_i	V_i*S_i
disk	7	0.012	0.084
network	6	0.025	0.150
CD	2	0.070	0.140
CPU	16	0.005	0.080
Total			0.454

What about Queuing?

- The many requests to the CPU and I/O devices will create queues
- The time for a device to service a request should include the waiting time.

- We will add the queuing time on Wednesday