Advanced Computer Architecture

Lecture No. 42

Reading Material

Patterson, D.A. and Hennessy, J.L. Chapter 8
Computer Architecture -A Quantitative Approach

Summary

- Introduction
- Performance of I/O Subsystems
- Loss System
- Single Server Model
- Little's Law
- Server Utilization
- Poisson distribution
- Benchmarks programs
- Asynchronous I/O and operating system

Introduction

Consider a producer-server model. A buffer (or queue) is present between them. Tasks are being received and when one task is finished (i.e. served) then the second task is taken up by the server. Now latency and the response time depend upon how many tasks are present in the queue and how quickly they are served. If there is no task, ahead in the queue the latency would be low and response time would be shorter.
Through put depends upon the average number of calls and the service time taken by a particular server.

Performance of I/O Subsystems

There are three methods to measure I/O subsystem performance:

- **Straight away calculations using execution time**
- **Simulation**
- **Queuing Theory**

Loss System

Loss system is a simple system having no buffer so it does not have any provision for the queuing. In a loss system, provision is time in term of how many switches we do need, then provide some redundancy how many individuals I/O controllers we do need, then how many CPUs are there. It is also called dimension of a loss system.

Delay System

This system provides additional facilities. If we find some call party busy, we can have provision of call waiting. If we have more than one call waiting, then once we finish the first call, we may receive the second call.

Single Server Model

Consider a black box. Suppose it represents an I/O controller. At the input, we have arrival of different tasks. As one task is done, we have a departure at the output. So in the black box, we have a server. Now if we expand and open-up the black box, we could see that incoming calls are coming into the buffer and the output of the buffer is connected to the server. This is an example of "single server model".

Little's Law

For a system with multiple independent requests for I/O service and input rate equal to output rate, we use Little's law to find the mean number of tasks in the system and Time sys such that

Mean number of tasks = Arrival Rate x Mean Response time
and
$Time_{sys} = Time_q + Time_s$
where
$Time_s$ = Average time to serve task
$Time_q$ = Average time per task in the queue
$Time_{sys}$ = Aver time /task
Arrival Rate = λ = Average number of arriving tasks
$Length_s$ = Average number of task in service
$Length_q$ = Average length of queue
and
$Length_{sys} = Length_q + Length_s$

Server Utilization

Server Utilization = Arrival Rate x $Time_q$

Server utilization is also called traffic intensity and its value must be between 0 and 1. Server utilization depends upon two parameters:

1. Arrival Rate
2. Average time required to serve each task

So, we can say that it depends on the I/O bandwidth and arrival rate of calls into the system.

Example 1

Suppose an I/O system with a single disk gets (on average) 100 I/O requests/second. Assume that average time for a disk to service an I/O request is 5ms. What is the utilization of the I/O system?

Solution

Time for an I/O request = 5ms
=0.005sec
Server utilization = 100 x 0.005
= 0.5

Poisson distribution

In order to calculate the response time of an I/O system, we make the following assumptions:

1. Arrival is random
2. System is memory less. It means that incoming calls are not correlated.

For characterize random events, according to above two assumptions, we use Poisson distribution:

Probability (k)= $(e^{-k} \times a^{k}) / k!$

a= Rate of events x Elapsed time
= Arrival rate x t

also

$$C^2 = \frac{\text{Variance}}{(\text{Arithmetic mean time})^2}$$

and

Average Residual Service Time = ½ x weighted mean time x $(1+C^2)$

Example 2

For the system of previous example having server utilization of 0.5, what is the mean number of I/O requests in the queue?

Solution

$$\text{Length}_q = \frac{(\text{Server utilization})^2}{(1 - \text{Server utilization})}$$

$\text{Length}_q = (0.5)^2 / (1-0.5) = 0.5$

Assumptions about Queuing Model

1. Poisson distribution is assumed
2. The system is in equilibrium
3. The length of the queue is infinity
4. The system has only one server

5. The server will start the next task after finishing the previous one.

Example 3

Suppose a processor sends 10 disks I/O per second, these requests are exponentially distributed, and the average service time of an older disk is 10ms. Answer the following questions:

- What is the number of requests in the queue?
- What is the average time a spent in the queue?
- What is the average response time for a disk request?

Solution

Average number of arriving tasks/second = 20
Average disk time = 10ms = 0.01sec
Sever utilization = 20 x 0.01=0.2
$Time_q$ = 10ms x 0.2/(1-0.2) = 2.5ms
Average response time = 2.5+10=22.5ms

M/M/m model of queuing theory
A system which has multiple servers is called M/M/m model.
The following formulas are used for M/M/m model:

$$\text{Utilization} = \frac{\text{Arrival Rate} \times Time_s}{N_s}$$

$$Length_s = \text{Arrival Rate} \times Time_q$$

$$Time_q = \frac{(Time_s \times (P_{tasks} >= N_s))}{N_s \times (1 - \text{utilization})}$$

$$Prob_{tasks} >= N_s = \frac{N_s \times \text{utilization}}{N_s! \times (1-\text{utilization})} \times Prob_{0tasks}$$

Example#4

Suppose instead of a new, faster disk, we add a second slow disk, and duplicate the data so that read can be serviced by either disk. Let's assume that the requests are all reads. Recalculate the answers to the earlier questions, this time using an M/M/m queue.
Solution

The average utilization of the two disks is given as;

$$\text{Server utilization} = \frac{\text{Arrival rate} \times \text{Time}_s}{N_s}$$
$$= (20 \times 0.01) / 2$$
$$= 0.1$$

$$\text{Prob}_{0\text{tasks}} = \left[1 + \frac{(2 \times \text{utilization})^2}{2! \times (1 - \text{utilization})} + \frac{(2 \times \text{utilization})^n}{n!} \right]^{-1}$$

$$\text{Prob}_{0\text{tasks}} = \left[1 + \frac{(2 \times 0.1)^2}{2! \times (1 - 0.1)} + (2 \times 0.1) \right]^{-1}$$

$$= (1 + .022 + 0.2)^{-1}$$

$$= 1.222^{-1}$$

$$\text{Prob}_{\text{tasks}} >= N_s = \frac{(2 \times \text{utilization})^2}{2! \times (1 - \text{utilization})} \times \text{Prob}_{0\text{tasks}}$$

$$= \frac{(2 \times 0.1)^2}{2! \times (1 - 0.1)} \times 1.222^{-1}$$

$$= 0.018$$

$$\text{Time}_q = \text{Time}_s \times \frac{\text{Prob}_{\text{tasks}} >= N_s}{N_s \times (1 - \text{utilization})}$$

$$= 0.01 \times 0.018 / (2 \times 0.9)$$
$$= 0.1\text{msec}$$

Average response time = 10msec + 0.1msec
= 10.01msec

Benchmarks programs
In order to measure the performance of real systems and to collect the values of parameters needed for prediction, Benchmark programs are used.
Types of Benchmark programs
Two types of benchmark programs are used:
TPC-C
SPEC

Asynchronous I/O and operating system

In order to improve the I/O performance, parallelism is used.
For this, two approaches are available:

- Synchronous I/O
- Asynchronous I/O

Synchronous I/O

In this approach, operating system requests data and switches to another process. Until the desire data arrived. Then the operating system switches back to the requesting process.

Asynchronous I/O

This model is of the process to continue after making a request and it is not blocked until it tries to read requested data.

Bus versus switches

Consider a LAN, using bus topology. If we replace the bus with a switch, the speed of the data transfer will be improved to a great extent.