ESTIMATING NETWORK ROTATIONAL LATENCY BASED ON INPUT/ OUTPUT SUBSYSTEM ARCHITECTURE

Akhigbe – Mudu Thursday Ehis

Department of Computer Science, Federal University of Agriculture Abeokuta. Nigeria.

ABSTRACT: Providing network with sufficient bandwidth is necessary, but not sufficient, step to ensuring good performance of a network application. If excessive network Latency is causing the application to spend a large amount of time waiting for responses, then the bandwidth may not be fully utilized and performance will suffer. Thus, an important aspect of data communication is to estimate the rotational latency cause by I/O subsystem. In this paper, we present a single model structure that can be used to represent complex I/O subsystems at varying levels. We developed some parameters for estimating the contention components of the effective service demands under a number of different assumptions and incorporating these parameters into network modeling. The result demonstrates real world effects of latency using the time to load a web page. This clearly shows that latency directly affects the way a user obtains data from the internet.

KEYWORDS: latency, contention, Rotation, Disks, Channel, Control Unit.

1. INTRODUCTION

Processor and primary memory technology has moved forward rapidly in recent years, but there has not been such comparable advances in the design of I/O subsystems. As a result, I/O subsystems are playing an increasingly critical role in computer system performance. It is against this background this paper is motivated. Internet data is packaged and transported in small pieces of data. The flow of these small pieces of data directly affects a user's internet experience. When data packets arrive in a smooth and timely manner the user sees a continuous flow of data, if data packets arrive with large and variable delays between packets, the user's experience is degraded. Latency is another element that contributes to network speed. The term latency refers to any of several kinds of delays typically incurred in processing of network data. Latency is a time delay between the moment something is initiated, and the moment one of its effects begins or becomes detectable. The word is derived from the fact that during the period of latency the effects of an action are latent meaning, "potential" or "not yet observed". Most people understand that it takes time for web pages to load and for emails to get from your outbox to the destination inbox and this is a form of latency. Therefore, latency is a time delay imparted by each element involved in the transmission of data [Spe91]. It is shown that in many environments the rotational latency and the Rotational Position Sensing (RPS) miss delays are the major contributors to a disk's basic service time. A sensitivity study using a simple analytical model shows that a reduction in these two components (both of which are related to the rotation of disks drives) has the impact in reducing the disk's basic service time and in turn produces the greatest improvement in overall performance.

1.1. Statement of the Problem

The architectural complexity of the subsystem we discuss here results from difficult compromises between cost and performances. At one extreme, requiring the CPU to monitor directly all phases of I/O activity would lead to poor performances (although low cost). At the other extreme, endowing each disk with sufficient intelligence to transform data in a fully independent manner would lead to high cost (although good performance). The obvious approach is to introduce some number of shared devices of varying intelligence (channels, controllers, string heads etc) on the path between the CPU and the disks. Disk with a feature called RPS allows the channel, control unit and head of string to be free during the latency period. The RPS disk contacts the control devices to establish an I/O path to transfer data when it is about to have the data under the heads [KRA13]. Again if any of the devices of the I/O path is busy at that instant, the disk has to wait a complete revolution until the data are in position again. The RPS reconnects delays sets in, because of lack of buffers in the disk. This may become too long, thereby degrading the disk performance

1.2. Input/output Subsystem Architecture

An I/O subsystem is a set of components responsible for controlling and executing I/O operations. Disk drives are connected to main memory via a series of devices that form the I/O interface. The complexity of an I/O interface varies as a function of its performance, reliability and cost. Figure 1, exhibits the structure and main components of a typical I/O Architecture.

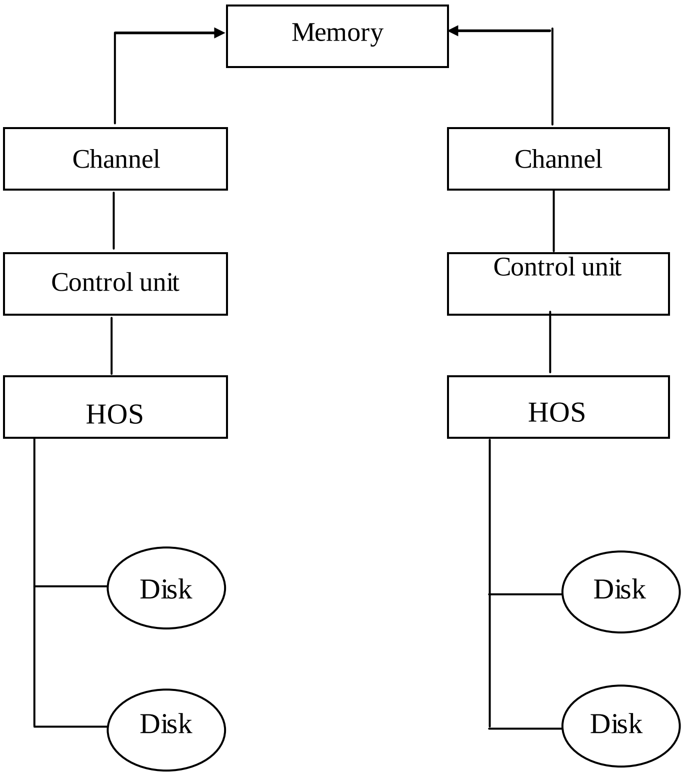

Figure 1: Typical Input/output Subsystem Architecture

Moving outwards the memory, the first component is a channel, which provides a communication path between memory and I/O devices. A channel is busy when it is transferring data. The next component is the control unit or disk controller. A control unit decodes the device specific I/O commands (e.g. seek and transfer) into control signals for the associated disks. It also makes the connection between many channels and many disks. A string of disks is attached to a device called head of string (HOS) which is responsible for controlling communications between control units and disks. The outermost components of the I/O subsystem are the disk unit, which can be viewed as a single server that services one request at a time. An I/O path is a physical connection between memory and an I/O device. A path is considered busy if any path of it is busy. To improve availability and performance, I/O subsystems have redundant paths from memory to I/O devices. In figure 1, the dashed lines indicate the existence of a string switch that allows a disk to be accessed by the two channels. The control devices shared by a number of disk drives, provides multiple pathways between memory and disks. The rest of the paper is structured as follows: section 2 describes the published related work. The methodology and the analysis model are included in section 3. Section 4 presents the implementation and the performance evaluation, while the last section summarizes the conclusion.

2. LITERATURE REVIEW

It is important to understand the very basic elements of networking to properly grasp the latency issue. The need to be able to handle thousands to millions of users on one cohesive, network, and thus the Transport Control Protocol/ Internet protocol (TCP/IP) networking model was developed [DRS12]. The key design feature of the TCP/IP networking model is the concept of encapsulation which is the idea of taking data and wrapping it in common container for shipping. The container that was developed is called the IP Datagram also known as an IP packet. The IP packet is a very simple thing: a header followed by data. The Header contains information used for routing the packet to the destination [Boj13]. The data can be any information which needs to be transported. The exact construct of the data portion of an IP packet is defined by the data protocol that is being carried. To understand exactly where Latency occurs, it is valuable to know how this most basic unit of networking data is built and transported. In IP networks such as the Internet, IP packets are forwarded from source to destination through a series of IP routers or switches that are interconnected by links such as circuits [BCB12]. The IP routers use the destination address in the IP header to determine the next router in the path from source to destination. The IP routers utilize routing algorithms to continuously update their decision about which router is the best one to get the packet to its destination. [Spe91] proposed the management of the wealth of internet traffic which includes delays (Latencies), caused by the routing and switching process. This refers to the amount of processing time for a router or switch to receive a packet, process it and transmit it on its way. [Tor10] proposed a sensitivity study using a simple analytical model to show a reduction in the two rotational disk components (both of which are related to the rotation of disk drives), have the ability to reduce the disk's basic service time and in turn produce the greatest improvement in overall performance. There are some limitations in these approaches. Several alternatives to reducing latency and RPS miss penalty are proposed and explored but their performances were analyzed using analytical queuing models.

3. METHODOLOGY

Parameterzing the I/O subsystems (the disks and channels) are complicated. The disk technology of the system requires that both disk and channel be held during rotational latency (the period during which the data is rotating to the read/write heads of the disk) and data transfer, while seeks could proceed at each disk independently of its channel. The basic disk service time consist of three parts: seek time, latency time and data transfer time. The seek time corresponds to the time required to move the read/write heads over the desired position. A disk unit is capable of doing this operation independently, which means that during the seek, control unit, string controller and channel are free to take on other activities. Once the read/write heads moved into place, the device has to wait until a particular sector of the disk comes under the heads and this time is referred to as the Latency. At this point, disk with a feature called "rotational Position Sensing" (RPS) allows the channel, control unit and Head of string to be free during the latency time. When an RPS disk is about to have the data under the heads, it contacts the control devices to establish an I/O path to transfer the data. If any of the devices of the I/O path is busy at that instant (transferring data on behalf of other disk units), the disk have to wait a complete revolution until the data are in position again. The RPS reconnects delays set in, from the lack of buffers in the disk and from the timing constraints of the operation. The RPS reconnects time may become too long; degrading the disk performance i.e. projected performance measures would be seriously in error.

3.1. Estimating Delays within I/O Subsystem

In case of RPS feature, the disk attempts to reconnect to the I/O path. If the channel / path is busy, the disk unit waits one full revolution until the data are under the heads and then again attempts to reconnect. This step is repeated until an attempt succeeds. Thus the waiting time in this step is an integral multiple of a full disk revolution time.

Let $P'(path.busy)$ denotes the probability that the I/O path for disk i is busy. $P'(path.busy)$ is also the probability that a reconnect fails and $(1-P_i)$ the probability of a successful reconnect. We can write that the probability of having k – RPS misses is given by $(1-\mathrm{P}_i)x\mathrm{P}_i(path.busy)^k$. Then the average number of misses is given by:

$$NRPS_i=\sum_{k=1}^{\infty}k[(1-\mathrm{P}_i)(path.busy)x\mathrm{P}_i(path.busy)]^k$$

$$=\frac{\mathrm{P}_i(path.busy)}{1-\mathrm{P}_i(path.busy)} \tag{1}$$

Multiplying the average number of misses reconnects by the rotation time of $disk_i$, we obtain the average RPS delay, DRPS

$$DRPS_i=NRPS_i xrotation_i \tag{2}$$

Let us assume that the channel is the major delay component of an I/O path, so that the delays caused by the other elements of the pathways are negligible. Therefore, we have

$$\mathrm{P}_i(path.busy)=\mathrm{P}_i'(channel.busy) \tag{3}$$

where $\mathrm{P}_i'(channel.busy)$ is the probability that $disk_i$ sees the channel busy.

The probability $\mathrm{P}_i'(channel.busy)$ refers to situations when the disk is not using the I/O path.

$$\mathrm{P}_i'(channel.busy)=\frac{\mathrm{P}_i'(channel.busy)}{disk_i} \tag{4}$$

From the definition of conditional probability $P(A/B)=P(A\cap B)/\mathrm{P}(B)$ From Eq. (4) we have

$$P'(channel.busy)=\mathrm{P}_i(channel.busy)\cap(disk_i not.transferring.data) \tag{5}$$

A system is considered busy when one or more customers are in the system, which is indicated by the utilization of the server. Thus, we have

$$\mathrm{P}_i'(channel.busy)=\frac{\left[\sum_{i=1}^{k}U_{ch}(k)\right]-U_i(transferring)}{1-U_i(transferring)} \tag{6}$$

where U_i indicates the utilization of $disk_i$, τ the monitoring period and C_0 the number of I/O operations carried out during τ. The product $U_i x\tau$ represents the time that $disk_i$ was busy. Also $\sum_{i=1}^{k}U_{ch}(k)$ represents the utilization of channel by all the k devices attached to it. When $disk_i$ is transferring data, it is using the channel, which means that the time $disk_i$ is dedicated to transferring data is equal to the time the channel is busy transferring data on behalf of $disk_i$. Thus, we can say that utilization of $disk_i$ due to transfers, $U_i(transferring)$ is equal to utilization of the channel due to the transfers of

$disk_i$, Letting $U_{ch} = \sum_{i=1}^{k} U_{ch}$ (k) and substituting Eq.(6) into equation (1), we have:

$$NRPS_i = \frac{U_{ch} \quad -U_{ch}}{1-U_{ch}}(i) \qquad (7)$$

3.2. Channel Contention

Let's consider the case of intelligent disks that are capable of performing seek and latency independently and have track buffers for queuing results. These disks do not suffer from RPS problems. However, a disk may still have to wait while the channels (or I/O bus) is busy transferring data of other disks. This kind of delay arises due to channel contention. The problem here is to estimate the delay. An approximate method for calculating the delay is to view the channel as an open single server, receiving requests from disks that want to transfer data. When $disk_i$ attempts to seize the channel to transfer data, it may find a number of requests from other disks waiting for the channel. In an open system with a single server, the average queue length seen by an arriving customer is $\frac{U}{(1-U)}$, where U is the system utilization. As the channel is being viewed as an open system, the average number of requests $(NCH)_i$ seen by $disk_i$ is given by: of

$$NCH_i = \frac{P_i'(channel.busy)}{1-P_i'(cahnnel.busy)} \qquad (8)$$

Considering that each disk can only have one I/O request at a time, $\text{P}_i'(channel.busy)$ represents the probability that the channel is busy given that $disk_i$ is not transferring data.

Therefore,

$$NCH_i = \frac{U_{ch} - U_{ch(i)}}{1-U_{ch}} \qquad (9)$$

Assuming that transfer is the average time for transferring data over the channel, we can express the average waiting time (TCH_i) of $disk_i$ due to contention:

$$TCH_i = NCH_i x.transfer \qquad (10)$$

3.3. Implementation

We use information that is typically provided by disk manufacturers, such as average seeks time, transfer rate and rotational speed. Using information from product specification will facilitates the process of obtaining input parameters for the model. The system under consideration is a Samsung system (product model: Z10). The architectural characteristics that we address and the modeling techniques that we develop, however, are equally applicable to systems of other manufacturers.

3.3.1. Example

Let's consider a database server with a processor and an I/O subsystem consisting of an I/O bus (i.e. the channel), a disk controller, and two disk drives (A and B). The server executes on the average of two transactions per second. A typical transaction requires 0.2sec of CPU and performs 8 input/output operations on disk A and 14 on disk B. The size of the block transferred in each operation is 0.5kbytes. The disks rotate at 3600rpm (rotations per minute), the advertised average seeks time is 15msec, the transfer rate is 2Mbytes/sec, and the controller overhead is 1msec. we want to calculate the average transaction response time for three models corresponding to situations that:

(1) Ignore contention with the I/O subsystem,

(2) consider content and assume tha the disk have the RPS features.

(3) Consider contention and assume that the disk have a track buffer.

Before we design an analytical open model to represent server performance, the first step is to determine the disk demands for each situation under analysis. The basic service time of a disk is:

$$S_b = seek + Latency + Trasnfer \qquad (11)$$

The latency is considered to be one half of the disk rotation period

$$\frac{1}{2} x \frac{1}{3600} = 8.3 m\sec \qquad (12)$$

The average transfer time for 512bytes equals 0.5/2 = 0.25msec.

Putting these values into equation (11) we have: $S_b = 15 + 8.3 + 0.25 = 23.55 m\sec$. The effective service time can then be expressed as:

$$S_{ef} = S_b controller.time + contention \qquad (13)$$
$$= 24.55 + contention$$

The problem now is to estimate contention for the three situations of the example. First we ignore contention and $S_{ef} = 24.55m\sec$. The service demands for disk A and disk B are 24.55 x 8 = 196 msec and 24.55 x 14 = 343.7 msec respectively.

For the RPS disks, we shall proceed to calculate $DRPS_i$, the average RPS delay for disk (i). To do that, we first need to estimate the average number of misses, NRPS(i). The channel utilization due to disk(i) is: $U_{ch}(i) = \chi_0 \times v_i \times transfer.time$

Thus, $u_{ch}(A) = 2 \times 8 \times 0.25 = 4\%$

$u_{ch}(B) = 2 \times 14 \times 0.25 = 7\%$ and the total channel utilization is $u_{ch} = 11\%$

Using Eq(1) we obtain

$$NRPS_A = \frac{0.11 - 0.04}{1 - 0.11} = 0.079$$

$$NRPS_B = \frac{0.11 - 0.07}{1 - 0.11} = 0.045$$

The average delays for the two disks are:

$DRPS_A = 0.079 \times 16.6 = 1.311m\sec$ and

$TCH_B = 0.045 \times 0.25 = 0.0113m\sec$

$DRPS_B = 0.045 \times 16.6 = 0.747m\sec$

With the RPS reconnect delays estimated, we are able to calculate the effective service time and consequently, the disk demands.

Thus, $D_A = (24.55 + 1.311) \times 8 = 206.89m\sec$ and

$D_B = (24.55 + 0.747) \times 14 = 354.16m\sec$

Disks with track buffer do not suffer from RPS problem, because they store the desired data into local buffers. Using Eq(7), we determine the average waiting time of a disk due to bus contention.

$TCH_A = 0.079 \times 0.25 = 0.0198m\sec$ and

$TCH_B = 0.045 \times 0.25 = 0.0113m\sec$.

The disk service demands for this case are as follows:

$D_A = (24.55 + 0.0198) \times 8 = 196.56m\sec$ and

$D_B = (24.55 + 0.0113) \times 14 = 343.86m\sec$

We have just discovered that in this case, transfer time is much smaller than rotation time; the delay caused by bus contention is insignificant when compared to the RPS delays.

3.4. Performance Evaluation

The spreading of net data over time reduces what is called the effective Bandwidth of link. Packets are still being transported at the same bit rate but high latency networks becomes noticeable to the user; it is taking much more time for all the web page packets to arrive. It is this spread over time behavior of high latency networks that becomes noticeable to the users and creates the impression that a link is not operating at a high speed. Figure 2 is a demonstration of real world effects of latency taking time to load a web page. This is a common activity which clearly shows users that latency directly affects the way a user obtains data from the internet. The following plots show the effects of latency on the time to load a web page.

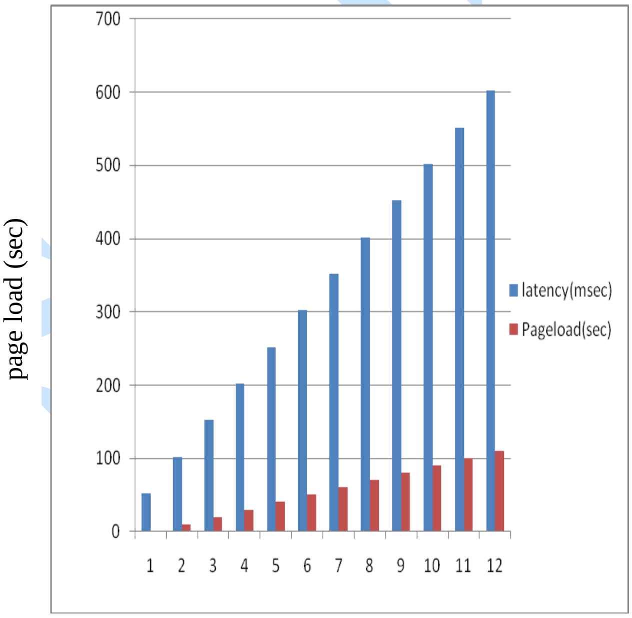

Figure 2: Latency (milliseconds)

4. CONCLUSION

In data communication, when data arrive in a smooth and timely manner the user sees a continuous flow of data, if data packets arrive with large and variable delays between packets, the user's experience is degraded. Latency is another element that contributes to network speed. We obtained information from product specifications to facilitate the process of obtaining input parameters for the model. In this paper, we presented a single model structure that can be used to represent complex I/O subsystems at varying levels. We developed some parameters for estimating the contention component of the effective service demands under a number of different assumptions. The simulation result shows that latency affects network bandwidth and network performance in general.

REFERENCES

[Boj13] **Florin Bojor** - *On Jachymski's Theorem,* Annals of the University of Crajova, Mathematics and Computer Science Series, Volume 40(10, 2013, pages 23 – 28, 2013

[BCB12] **Partha Sarathi Banerjee, J. Paul Choudhury, S. R. Bhadrachaudhuri** - *A framework for Selecting the most Reliable Path in a Computer Network Using Particle Swarm Optimization Based on Fuzzy Logic.* International Journal of Computer Applications (0975 – 8887), Volume 45, No. 8, 2012

[DRS12] **Sanjay Kumar Dubey, Ajay Rana, Arun Sharma** - *Usability Evaluation of Object Oriented Software System Using Fuzzy Logic Approach*, International Journal of Computer Applications (0975 – 8887), Volume 43, No.19, April 2012

[KRA13] **L. Kirichenro, T. Radivilova, Abed Saif Algehawli** - *Mathematical Simulation of Self – Semilar Network Traffic with Aimed Parameters.* Annals Computer Science series, 11th Tome, !st Fasc, pages 17 – 22, 2013

[Spe91] **W. Spencer** - *Improving disk performance via Latency reduction, IEEE Transactions on Computers Archive.* Vol. 40, Issue 1, pages 22 – 30, Jan. 1991

[Tor10] **Daniel Turull Torrents** - *Open Source Traffic Analyzer,* Master of Science Thesis. Stockholm, Sweden 2010, KTH Information and Communication Technology.