N93-30477

Analysis of Cache for Streaming Tape Drive

V. Chinnaswamy
8 Quail Hollow Road
Westboro, MA 01581

S28-35
159118
P, 12

1 Introduction

A tape subsystem consists of a controller and a tape drive. Tapes are used for backup, data interchange and software distribution. This paper is concerned only with the backup operation. During a backup operation, data is read from disk, processed in CPU and then sent to tape. The processing speeds of a disk subsystem, CPU and a tape subsystem are likely to be different. A powerful CPU can read data from a fast disk, process it and supply the data to the tape subsystem at a faster rate than the tape subsystem can handle. On the other hand, a slow disk drive and a slow CPU may not be able to supply data fast enough to keep a tape drive busy all the time. The backup process may supply data to tape drive in bursts. Each burst may be followed by an idle period. Depending on the nature of the file distribution in the disk, the input stream to the tape subsystem may vary significantly during backup. To compensate for these differences and optimize the utilization of a tape subsystem, a cache or buffer is introduced in the tape controller.

Most of the tape drives today are streaming tape drives. A streaming tape drive goes into reposition when there is no data from the controller. Once the drive goes into reposition, the controller can receive data, but it cannot supply data to the tape drive until the drive completes its reposition. This reposition time may vary from several milliseconds to a few seconds depending on the technology of the drive. A controller can also receive data from the host and send data to the tape drive at the same time.

This paper investigates the relationship of cache size, host transfer rate, drive transfer rate, reposition and ramp up times for optimal performance of the tape subsystem. Formulas developed here will also show the advantages of cache watermarks to increase the streaming time of the tape drive, maximum loss due to insufficient cache trade offs between cache and reposition times and the effectiveness of cache on a streaming tape drive due to idle times or interruptions due in host transfers.

In Section 2, several mathematical formulas are developed to predict the performance of the tape drive. Some examples are given in Section 3 illustrating the usefulness of these formulas. Finally, a summary and some conclusions are provided in Section 4.

2 Mathematical Analysis

The performance of a tape subsystem depends on several variables and their relationships. In this section, several formulas are developed for the throughput of the tape drive.

Let

- λ denote the host transfer rate,
- μ, the drive transfer rate,
- C, the cache size.

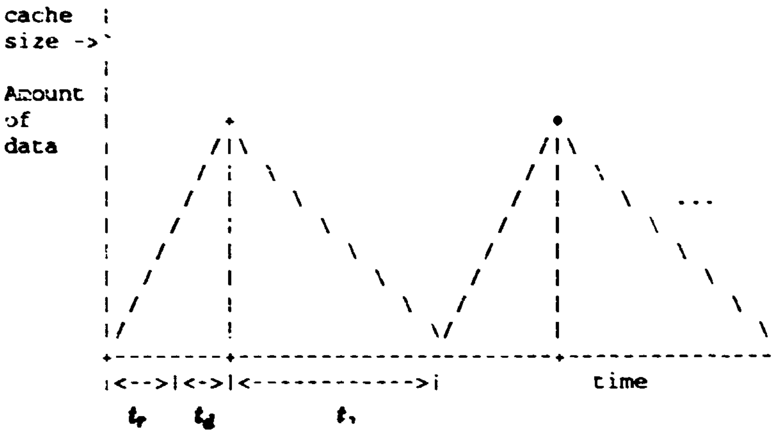
- t_r, the reposition time,
- t_d, the ramp up time delay to request,
- t_s, the streaming time before next reposition.

Any other variables of interest will be defined as needed. For now, refer to $P = t_r + t_d + t_s$ as a period. All the throughput numbers will be in kilobytes per second. All times will be in seconds.

2.1 Host transfer rate < the drive transfer rate

case i: $\lambda < \mu, \lambda(t_r + t_d) < C$, no idle time in host transfer,

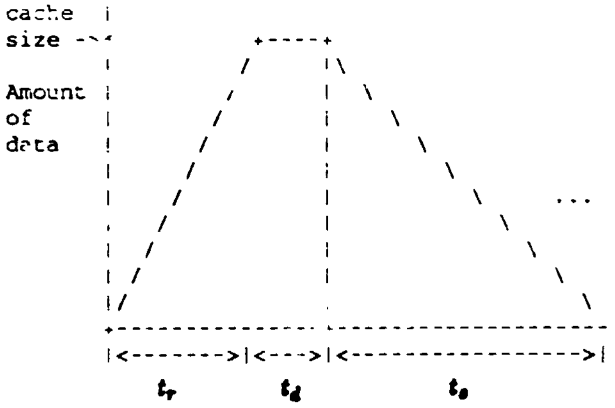
i.e., cache does not get filled up during reposition and ramp up time.



t_s denotes the drive streaming time. Each period repeats itself until the whole backup operation is over. So throughput can be calculated from just one period.

case ii: $\lambda < \mu, \lambda(t_r + t_d) > C$, no idle time in host transfer,

i.e., cache gets filled up during reposition and ramp up time.



When cache gets full and the drive is in reposition, host transfer gets blocked. When this happens bandwidth is lost. There is no idle time in host transfer except during this blocking time. The above two cases will be analyzed first before getting into several other cases.

Analyzing the figures for one period, we get the following relationships:

$$\lambda(t_r + t_d + t_e) = \mu t_e \quad if\ \lambda(t_r + t_d) \le C$$

$$C + \lambda t_e = \mu t_e \quad if\ \lambda(t_r + t_d) > C$$

From these two equations, we can get the value of t_e

$$t_e = \begin{cases} \frac{\lambda(t_r+t_d)}{(\mu-\lambda)} & if\ \lambda(t_r + t_d) \le C \\ \frac{C}{(\mu-\lambda)} & if\ \lambda(t_r + t_d) > C \end{cases}$$

Since the process repeats itself for each period, the effective throughput, T, of the tape subsystem can be calculated from one period.

$$T = \frac{Total\ Data\ Transferred\ by\ Drive}{Total\ Time} = \frac{\mu t_e}{(t_r + t_d + t_e)} = \frac{\mu}{1 + \frac{(t_r+t_d)}{t_e}}$$

We may often refer to T as an approximate throughput since we are neglecting the initial time due to label checking, track turn around time, etc. However, these times would become negligible when we are considering several hours of backup time.

Using the conditions above, we get

$$T = \begin{cases} \lambda & if\ \lambda(t_r + t_d) \le C \\ \frac{\mu}{1+\frac{(t_r+t_d)(\mu-\lambda)}{C}} & if\ \lambda(t_r + t_d) > C \end{cases}$$

2.1.1 Maximum loss in effective throughput

When $\lambda < \mu$ and $\lambda(t_r + t_d) > C$, the host transfer is blocked when the drive is in reposition. In this case, there is a loss in throughput due to insufficient cache.

The loss in throughput due to insufficient cache is given by

$$L = \lambda - \frac{\mu}{1 + \frac{(t_r+t_d)(\mu-\lambda)}{C}}$$

Differentiating with respect to λ, we can prove that the maximum loss occurs when

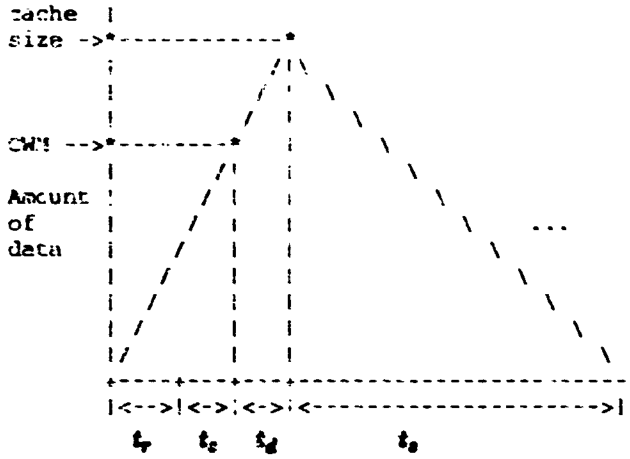
$$\lambda = (\frac{C}{t} + \mu) - \sqrt{\frac{C * \mu}{t}}$$

where $t = t_r + t_d$. For $C = 512$, $t = 1.35$, and $\mu = 800$, the maximum loss occurs when $\lambda = 628$. When $\lambda = 628$ KB/sec, we get only a throughput of 550 KB/sec, a loss of 78 KB/sec.

2.1.2 Cache Watermarks

When $\lambda(t_r + t_d) < C$, we migh' '' ink of introducing a watermark level at $C - \lambda t_d$ such that we fill up the cache before th ıpe drive starts transferring data.

When the controller tells the drive to start writing data, the drive does not start writing data immediately. There is a ramp up delay in its response time. This time is not negligible for some drives. Suppose the ramp time is .5 seconds. If the drive is told to transfer data when the cache is 100 percent full, the host transfer will be blocked for 500 milliseconds.



In this case, we have

$$\lambda(t_r + t_c + t_d + t_s) = \mu t_s \text{ if } \lambda(t_r + t_d) < C$$

where t_c is the additional wait time to bring the data in cache to the watermark level.

There is no point in setting a watermark if $\lambda(t_r + t_d) \geq C$.

Solving for t_s, we get

$$t_s = \frac{\lambda(t_r + t_s + t_d)}{(\mu - \lambda)}$$

Throughput = $T = \frac{\mu t_s}{(t_r+t_c+t_d+t_s)} = \frac{\mu}{1+\frac{(t_r+t_c+t_d)}{t_s}}$

Using the value of t_s, we get

$$T = \frac{\mu}{1 + \frac{(\mu-\lambda)}{\lambda}} = \lambda$$

Introducing a cache watermark has not changed the throughput. But the streaming time has increased (and consequently the number of repositions during a given time has decreased) since

$$t_s = \frac{\lambda(t_r + t_c + t_d)}{(\mu - \lambda)} > \frac{\lambda(t_r + t_d)}{(\mu - \lambda)}$$

Given the total time or total amount of data, we can easily calculate t' ı number of repositions saved by using the cache watermark. If increased number of repositions causes any reliability concerns, it is worth considering introducing cache watermarks when $\lambda(t_r+t_d) < C$. When $\lambda(t_r + t_d) > C$, there is no point in introducing a cache watermark. It does not change the throughput.

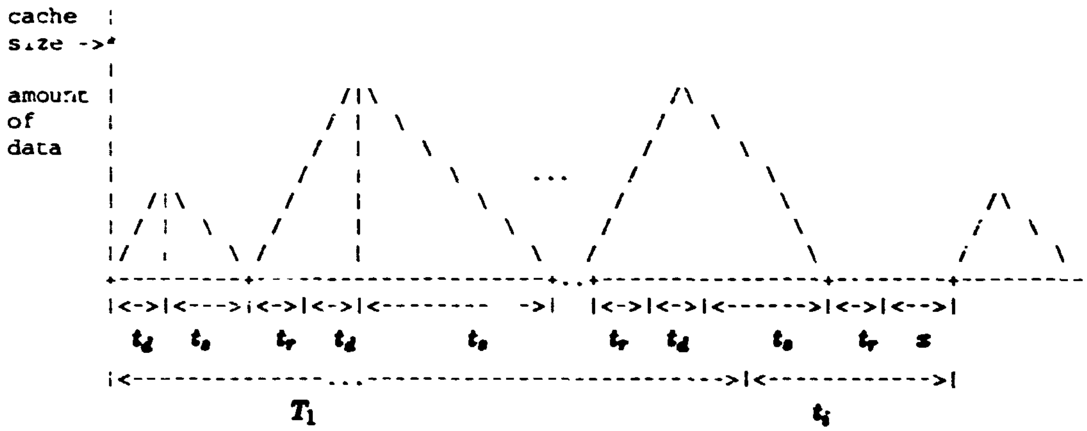
2.1.3 Host transmission has idle periods

Let

- T_1 be the continuous host transfer time
- t_i be the idle period before the next transmission.

We will assume that these times are contants and do not vary from period to period.

case iii: $\lambda < \mu, \lambda(t_r + t_d) < C, t_i > \frac{C}{\mu} + t_r$

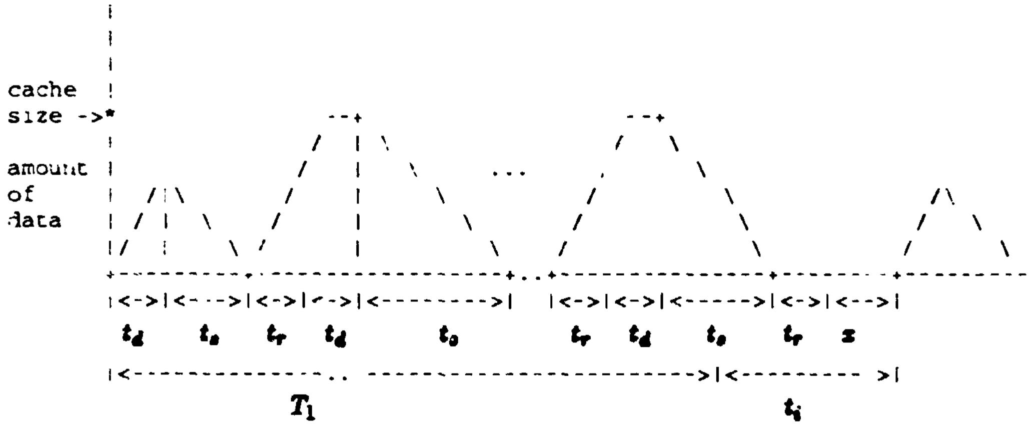


The approximate effective throughput is given by

$$T \approx \frac{T_1 \lambda}{T_1 + t_i}$$

The results are also true for the case $0 < t_i < \frac{C}{\mu} + t_r$

case iv: $\lambda < \mu, \lambda(t_r + t_d) > C, t_i > \frac{C}{\mu} + t_r$



In this case, the host transfer is blocked when the cache gets full. The approximate effective throughput is given by

$$T = \frac{[T_1 - n(t_r + t_d - \frac{C}{\lambda})]\lambda}{T_1 + t_i}$$

where n is given by

$$n = \lfloor \frac{T_1 - \frac{\lambda t_d}{\mu - \lambda}}{t_r + t_d + \frac{C}{\mu - \lambda}} \rfloor$$

If $[T_1 - n(t_r + t_d + \frac{C}{\mu-\lambda})] > (2 + \frac{\lambda}{\mu-\lambda})t_d + t_r$

then n = n + 1.

The results are also true for the case $0 < t_i < \frac{C}{\mu} + t_r$

2.2 Host Transfer Rate > Drive Transfer Rate

In this section, we will analyze all cases arising from the condition when host transfer rate exceeds the transfer rate of the drive.

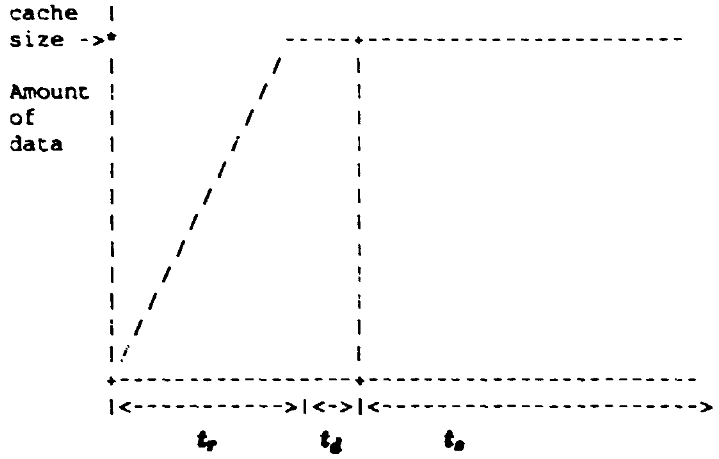
2.2.1 No Idle Time in Host Transfer

case v: $\lambda > \mu$, $\lambda(t_r + t_d) < C$, no idle time in host transfer.

```
cache  |
size ->*                  --------------------
       |                 /
Amount |                /
of     |               /
data   |              /
       |             /|
       |            / |
       |           /  |
       |          /   |
       |         /    |
       |        /     |
       |       /      |
       |      /       |
       +--------+------------------------------
       |<-->|<->|<------------------------------>
          t_r  t_d         t_s
```

In this case, input is blocked as soon as cache is full. Input rate will be limited to the output rate. Effective throughput of the tape drive is the maximum throughput capacity of the tape drive. Cache has no significant impact.

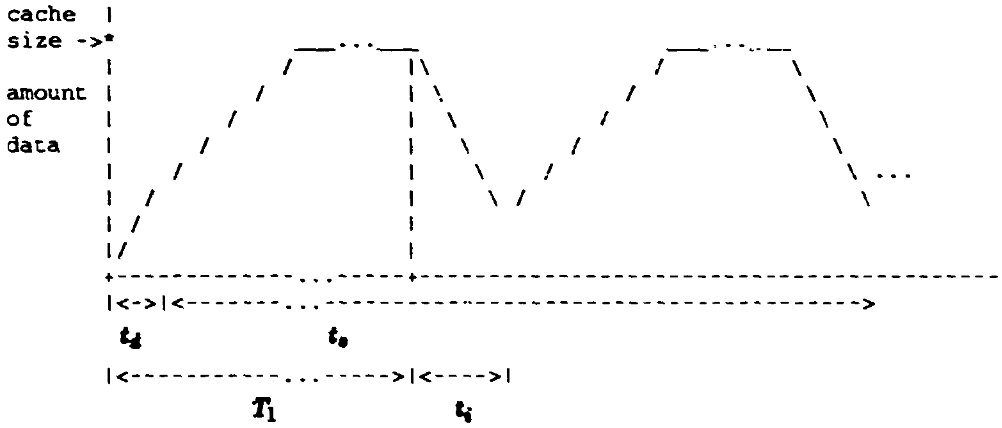
case vi: $\lambda > \mu$, $\lambda(t_r + t_d) > C$, no idle time in host transfer.



In this case, input is blocked as soon as cache is full. There is no input or output transmission for some period. This is a lost bandwidth. After this no transmission period, input rate will be limited to the output rate. Cache again has no impact.

2.2.2 Host transmission has idle periods

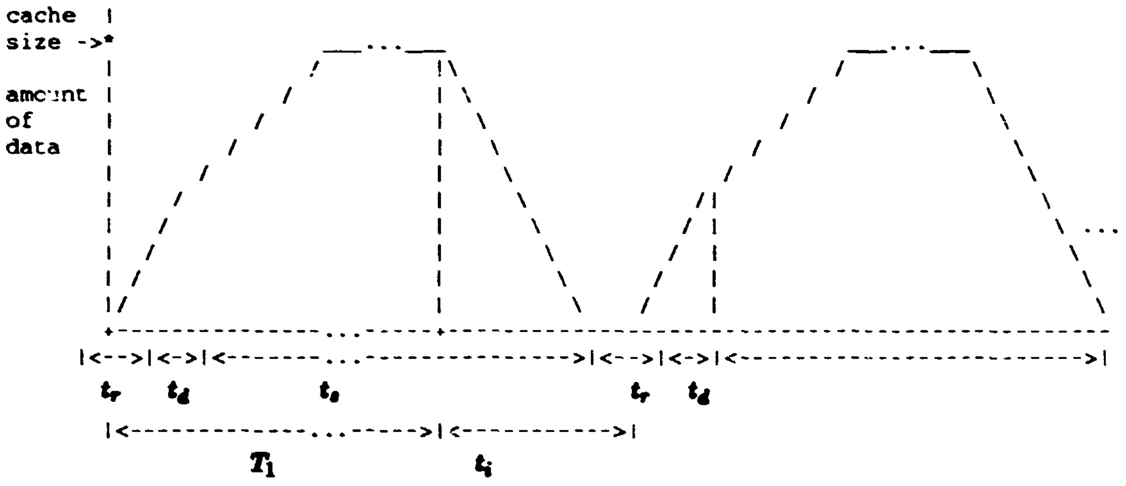
case vii: $\lambda > \mu, \lambda(t_r + t_d) < C, t_i < \frac{C}{\mu}$ or $\lambda > \mu, \lambda(t_r + t_d) > C, t_i < \frac{C}{\mu}$



In this case, the input transmission begins before the drive empties the cache. The drive streams all the time. The effective throughput is approximately the same as the drive transfer rate.

case viii: $\lambda > \mu, \lambda(t_r + t_d) < C, \frac{C}{\mu} < t_i < \frac{C}{\mu} + t_r$ or $\lambda > \mu, \lambda(t_r + t_d) > C, \frac{C}{\mu} < t_i < \frac{C}{\mu} + t_r$

i.e., the input transmission begins after the drive empties the cache, but before the drive completes its reposition.



$$T = \frac{(T_1 + t_i - \frac{C}{\mu} - t_r - t_d)\mu + C}{(T_1 + t_i)}$$

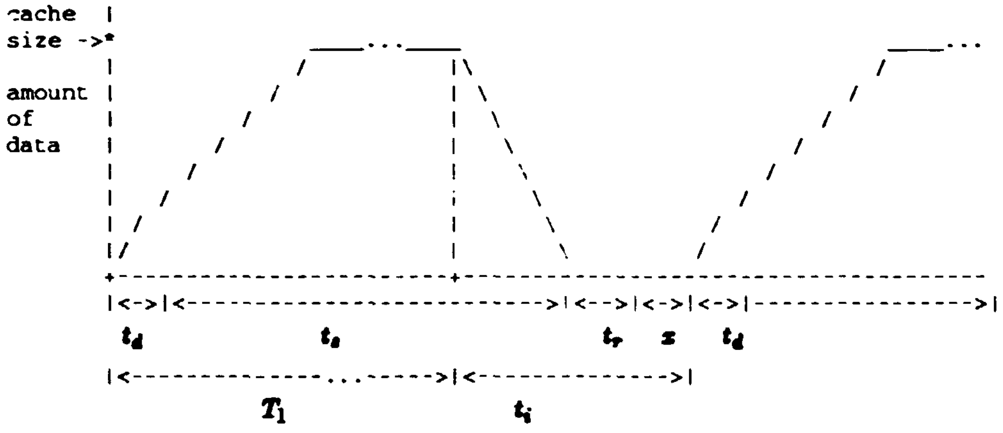
$$= \frac{(T_1 + t_i - t_r - t_d)\mu}{(T_1 + t_i)}$$

$$= (1 - \frac{t_r + t_d}{T_1 + t_i})\mu$$

The approximate effective throughput is less than the drive transfer rate. How much less will depend on the reposition time and idle time.

case ix: $\lambda > \mu, \lambda(t_r + t_d) < C, t_i > \frac{C}{\mu} + t_r$ or $\lambda > \mu, \lambda(t_r + t_d) > C, t_i > \frac{C}{\mu} + t_r$

The idle period is longer. The drive empties cache, completes reposition and then waits for data.



The effective throughput for one period is given by

$$T = \frac{(T_1 - t_d)\mu + C}{T_1 + t_i}$$

$$= \frac{(T_1 + t_i)\mu - t_i\mu + C - \mu t_d}{T_1 + t_i}$$

$$= (1 - \frac{t_i + t_d - \frac{C}{\mu}}{T_1 + t_i})\mu$$

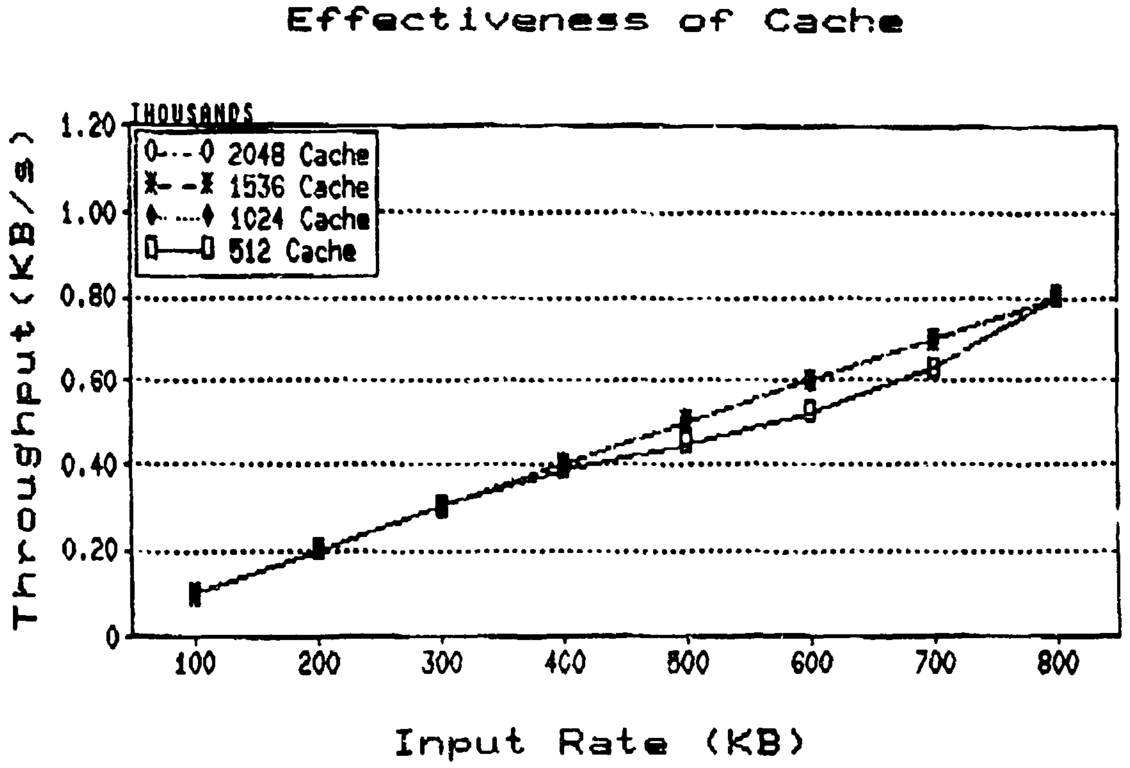
$$= (1 - \frac{t_r + t_d + x}{T_1 + t_i})\mu$$

The approximate effective throughput is less than the drive transfer rate. How much less will depend on the reposition time, wait time and idle time.

3 Some Illustrative Examples

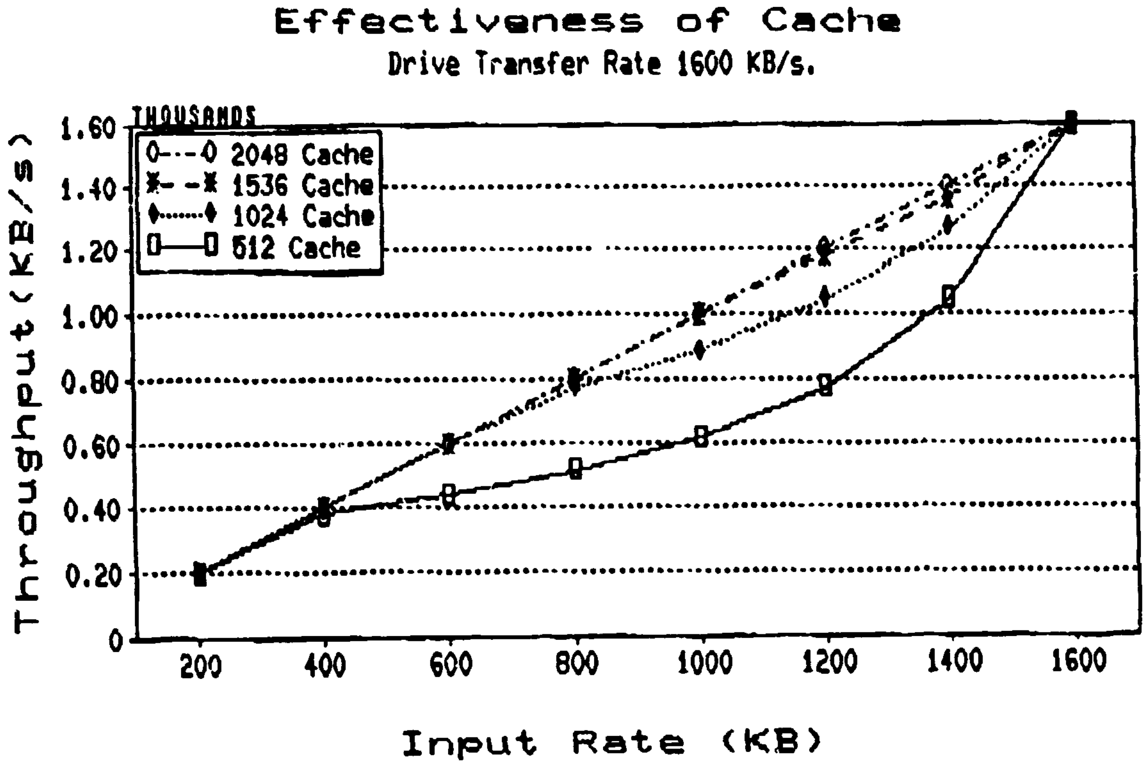
Suppose we have a drive with $\mu = 800$, $t_r = 1.0$ second, $t_d = 0.350$ second, $C = 512$ KB. For continuous host transfer and for all $\lambda < \mu$, the graph in Figure 1 gives the throughput for different cache sizes. We lose some throughput with 512 KB cache. 1024 KB cache gives better performance than 512 KB cache. More than 1 MB cache seems to be a waste.

Figure 1: Cache Sizes and Throughput



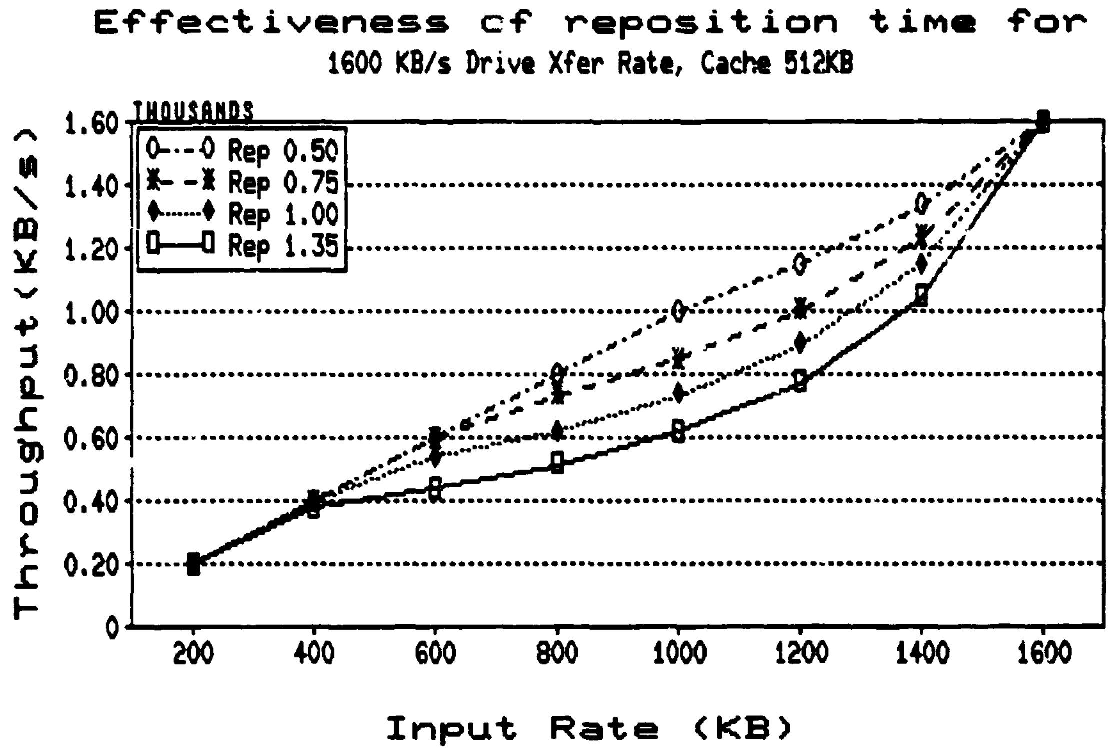
Let us consider the effect of increasing only the tape drive speed, i.e., $\mu = 1600$, $t_r = 1.0$ second, $t_d = 0.350$ second. Figure 2 shows the performance for various cache sizes. For all $\lambda < \mu$, increasing the drive transfer rate will decrease the performance of the system unless there is an increase in cache size. A cache size of 2 MB is needed when the drive transfer rate is increased to 1600 KB/sec.

Figure 2: Performance for Different Cache Sizes



A comparison of Figure 1 and Figure 2 shows that increasing the transfer rate of the tape drive without a comparable increase in cache size and/or decrease in reposition time has a negative impact in the performance for certain range of input values. The throughput can be increased by reducing the ıeposition and ramp up time instead of increasing the cache size.

Figure 3: Performance for different reposition times



4 Summary of Results and Conclusions

case i: $\lambda < \mu, \lambda(t_r + t_d) < C$, no idle time in host transfer $\Longrightarrow T \approx \lambda$

case ii: $\lambda < \mu, \lambda(t_r + t_d) > C$, no idle time in host transfer $\Longrightarrow T \approx \frac{\mu}{1+\frac{(t_r+t_d)(\mu-\lambda)}{C}}$

case iii: $\lambda < \mu, \lambda(t_r + t_d) < C, t_i > \frac{C}{\mu} + t_r \Longrightarrow T \approx \frac{T_1 \lambda}{T_1 + t_i}$

case iv: $\lambda < \mu, \lambda(t_r + t_d) > C, t_i > \frac{C}{\mu} + t_r \Longrightarrow T \approx \frac{[T_1 - n(t_r + t_d - \frac{C}{\lambda})]\lambda}{T_1 + t_i}$

case v: $\lambda > \mu, \lambda(t_r + t_d) < C$, no idle time in host transfer $\Longrightarrow T \approx \mu$

case vi: $\lambda > \mu, \lambda(t_r + t_d) > C$, no idle time in host transfer $\Longrightarrow T \approx \mu$

case vii: $\lambda > \mu, \lambda(t_r + t_d) < C, t_i < \frac{C}{\mu} \Longrightarrow T \approx \mu$

case viii: $\lambda > \mu, \lambda(t_r + t_d) < C, \frac{C}{\mu} < t_i < \frac{C}{\mu} + t_r \Longrightarrow T \approx (1 - \frac{t_r + t_d}{T_1 + t_i})\mu$

case ix: $\lambda > \mu, \lambda(t_r + t_d) < C, t_i > \frac{C}{\mu} + t_r \Longrightarrow T \approx (1 - \frac{t_r + t_d + s}{T_1 + t_i})\mu$

case x: $\lambda > \mu, \lambda(t_r + t_d) > C, t_i < \frac{C}{\mu} \Longrightarrow T \approx \mu$

case xi: $\lambda > \mu, \lambda(t_r + t_d) > C, \frac{C}{\mu} < t_i < \frac{C}{\mu} + t_r \Longrightarrow T \approx (1 - \frac{t_r + t_d}{T_1 + t_i})\mu$

case xii: $\lambda > \mu, \lambda(t_r + t_d) > C, t_i > \frac{C}{\mu} + t_r \Longrightarrow T \approx (1 - \frac{t_r + t_d + s}{T_1 + t_i})\mu$

When the host transfer rate is less than the drive transfer rate and if cache doesn't get filled up during reposition, the throughput rate would be the same as the host transfer rate. When the host transfer rate exceeds the drive transfer rate and either the host transfer has no idle time or the idle time is less than the time to empty cache, the throughput would be the same as the drive transfer rate. In all other cases, we lose throughput. The amount of loss would depend on the parameter values and their relationships.

In case ii, we lose throughput either because we have insufficient cache or the reposition time is high.

In cases iii, viii, ix, xi, and xii, we lose throughput because of idle time from host transfer. When there is an idle period, t_i, the tape drive

- will stream if $t_i < \frac{C}{\mu}$.
- will not stream if $t_i \geq \frac{C}{\mu}$.

In case iv, we lose throughput due to both idle time and insufficient cache.

These formulas are helpful to understand the behavior of the new tape subsystems when there are changes to any of the parameter values. They also predict the backup throughputs for any specified parameter values.