The Five-Minute Rule

Michael Whittaker

Abstract

Disk bandwidth costs money. RAM costs money. If I have a piece of memory, does it cost less money to store it on disk or in RAM? Jim Gray, Franco Putzolu, and Goetz Graefe initially answered this question with the *five-minute rule* in 1987 [3] and then again ten [2] and twenty [1] years later. In this paper, we explain the five-minute rule in plain English (Section 1) and then use it to make some pretty graphs (Section 2).

1 The Five-Minute Rule

A disk is a big long piece of ~~spaghetti~~ memory segmented into pages. The disk provides an interface that allows you to read it one page at a time. No more, no less. You want to read less than a page? Too bad. You can't. How big are the pages? They could be 4096 bytes or 1024 bytes or 42 bytes; it doesn't much matter to us. From now on, we'll denote the number of bytes in a page as PAGESIZE (e.g. 4096, 2048, 42).

When it comes to disks, faster is better. That is, we want our disks to be able to read as many pages per second as possible. Unfortunately, speed ain't cheap; we'll have to pay for throughput. A disk that can read 100 pages per second will cost more than one that can only read 1 page per second. We'll assume each page per second of throughput costs some fixed amount ACCESSPRICE (e.g. $2000).

Like disk, RAM is a big long piece of ~~spaghetti~~ memory. Unlike disk, RAM isn't divided into pages. We can read and write as much or as little data into RAM as we'd like. Want to read 4096 bytes from RAM? No problem. How about 2048 bytes? Fine by me. 42 bytes? Go ahead! Moreover, when it comes to RAM, bigger is better. Unfortunately, size ain't cheap; we'll have to pay for every byte of RAM. Let's assume that each byte of RAM costs some fixed amount RAMPRICE (e.g. $5).

Now imagine we have some object we'd like to store somewhere. Every so often, we read the object. For example, maybe the object is some 4 byte integer that we read every second. Or, maybe it's a 10 MB movie that we ready every 6 hours. Let MEMSIZE be the size of the object in bytes (e.g. 4, 10485760), and let ACCESSPERIOD be the number of seconds between each access (e.g. 1, 21600).

We can store this object on disk or in RAM. Let's look at how much both of these alternatives are going to cost us.

- *On Disk.* If our object is MEMSIZE bytes big and there are PAGESIZE bytes per page, then we'll have to store the object on $\lceil \frac{\text{MemSize}}{\text{PageSize}} \rceil$ pages. If we access the object every

$\textsc{AccessPeriod}$ seconds, then reading the object requires $\frac{\lceil \frac{\textsc{MemSize}}{\textsc{PageSize}} \rceil}{\textsc{AccessPeriod}}$ pages per second of bandwidth. With each page per second of bandwidth costing $\textsc{AccessPrice}$ dollars, storing the object on disk is going to cost, in dollars:

$$\frac{\lceil \frac{\textsc{MemSize}}{\textsc{PageSize}} \rceil}{\textsc{AccessPeriod}} \times \textsc{AccessPrice}$$

- *In RAM.* Our object is $\textsc{MemSize}$ bytes and each byte of RAM costs $\textsc{RamPrice}$ dollars, so storing the object in RAM costs, in dollars:

$$\textsc{RamPrice} \times \textsc{MemSize}$$

We want to store an object in RAM whenever it's cheaper to do so. That is, when

$$\frac{\lceil \frac{\textsc{MemSize}}{\textsc{PageSize}} \rceil}{\textsc{AccessPeriod}} \times \textsc{AccessPrice} \geq \textsc{RamPrice} \times \textsc{MemSize}$$

Solving for $\textsc{AccessPeriod}$, we get:

$$\textsc{AccessPeriod} \leq \frac{\lceil \frac{\textsc{MemSize}}{\textsc{PageSize}} \rceil \times \textsc{AccessPrice}}{\textsc{RamPrice} \times \textsc{MemSize}}$$

In words, given a disk with $\textsc{PageSize}$ bytes per page and $\textsc{AccessPrice}$ dollars per page per second of bandwidth, a RAM with $\textsc{RamPrice}$ dollars per byte, and a memory object that's $\textsc{MemSize}$ bytes big, we'll store the object in RAM whenever we access it every $\frac{\lceil \frac{\textsc{MemSize}}{\textsc{PageSize}} \rceil \times \textsc{AccessPrice}}{\textsc{RamPrice} \times \textsc{MemSize}}$ seconds or fewer.

Gray et al. substituted 1987 era values for $\textsc{PageSize}$, $\textsc{AccessPrice}$, $\textsc{RamPrice}$, and $\textsc{MemSize}$ and found $\textsc{AccessPeriod}$ to be around five minutes. Hence, the five-minute rule.

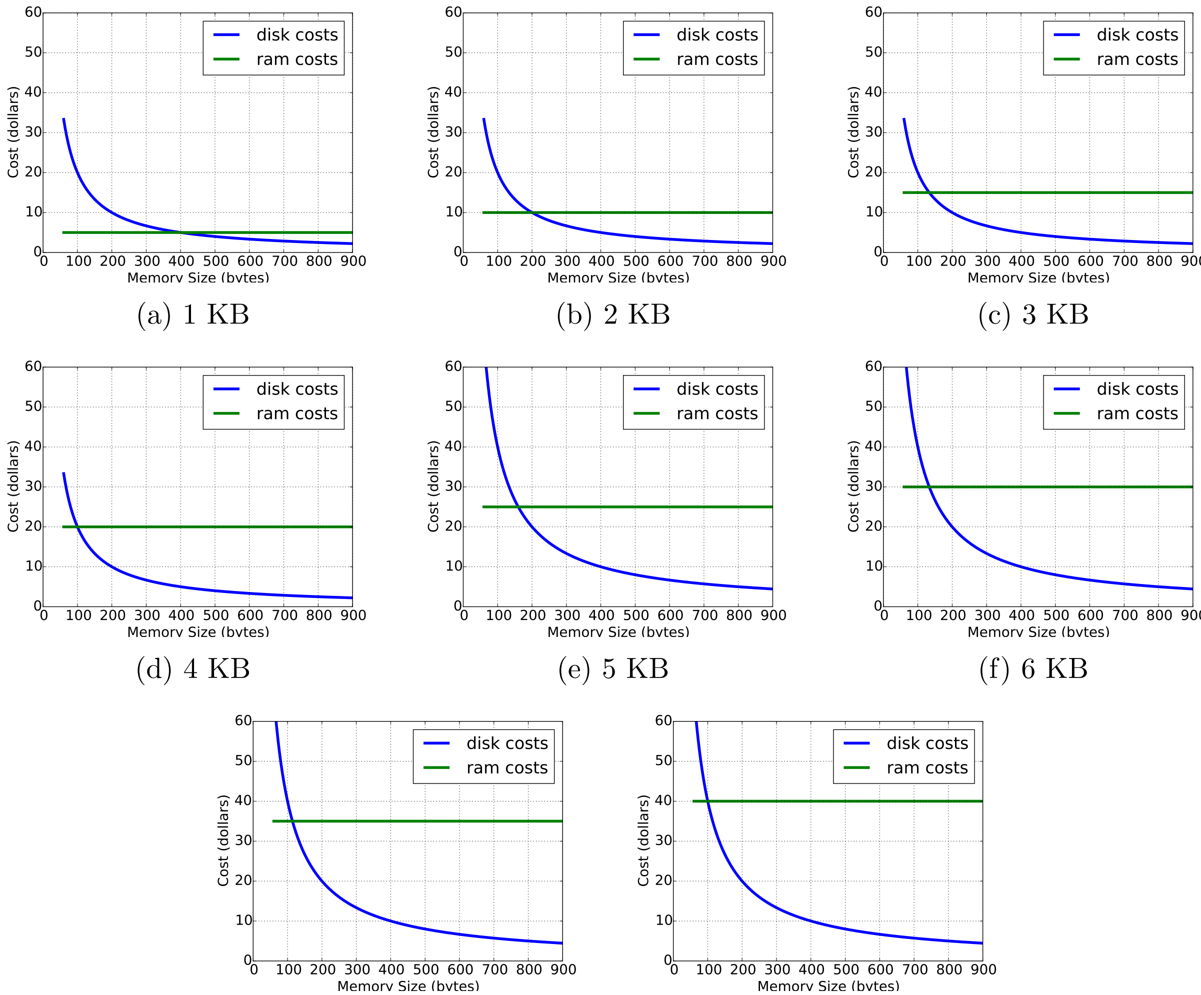
2 Some Pretty Graphs

We can use the five-minute rule formula to generate some nice graphs. Letting

- $\textsc{PageSize} = 4$ KB,
- $\textsc{AccessPrice} = \frac{\$2000}{page/second}$, and
- $\textsc{RamPrice} = \frac{\$5}{KB}$,

we can graph the cost of storing objects of various sizes with various access periods on disk and in RAM. This is shown in Figure 1. The point where the disk cost and RAM cost intersect is called the *break even point*. If the object is accessed less frequently than the break even point, then it should be stored in RAM; otherwise, it should be stored on disk.

Furthermore, given disks and RAMs with various values of $\textsc{PageSize}$, $\textsc{AccessPrice}$, and $\textsc{RamPrice}$, we can graph the break even point for objects of various sizes. This is given in Figure 2.



(a) 1 KB

(b) 2 KB

(c) 3 KB

(d) 4 KB

(e) 5 KB

(f) 6 KB

(g) 7 KB

(h) 8 KB

Figure 1: Disk and RAM costs

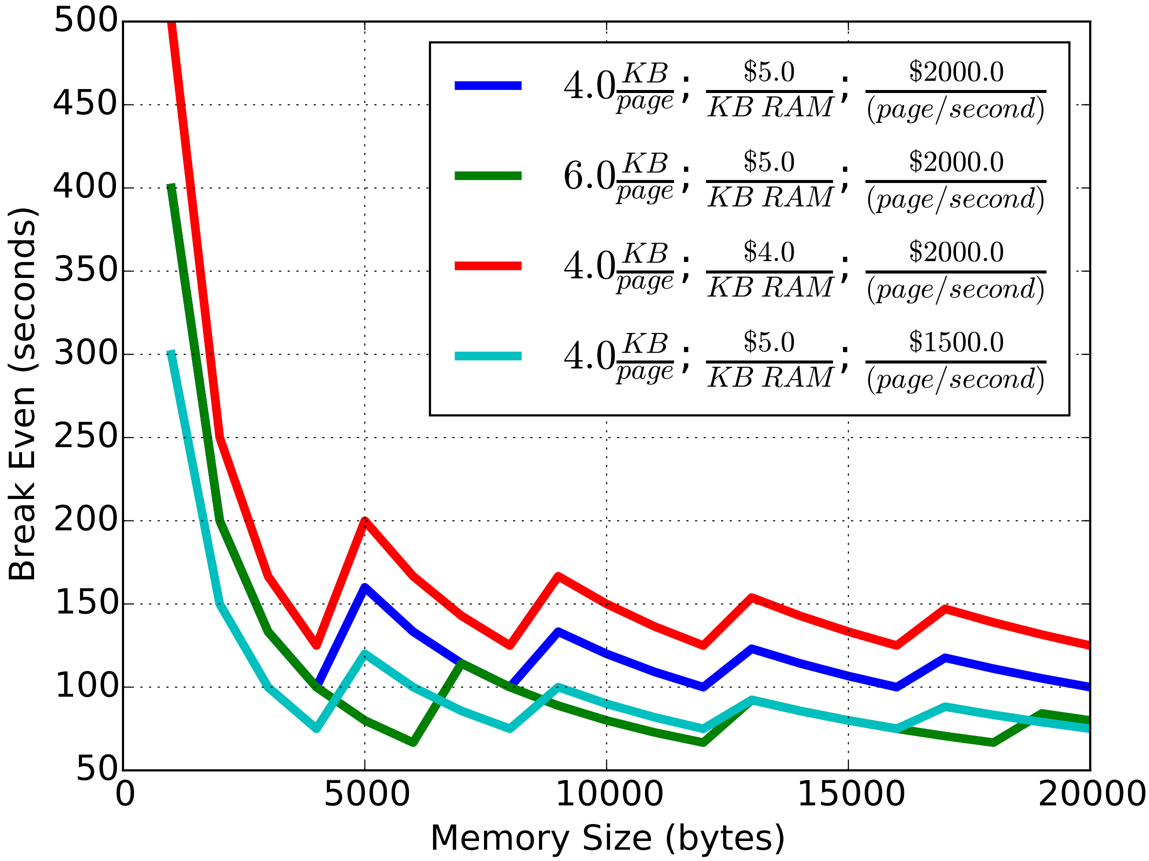


Figure 2: Break even points.

References

[1] Goetz Graefe. The five-minute rule twenty years later, and how flash memory changes the rules. In *Proceedings of the 3rd international workshop on Data management on new hardware*, page 6. ACM, 2007.

[2] Jim Gray and Goetz Graefe. The five-minute rule ten years later, and other computer storage rules of thumb. *arXiv preprint cs/9809005*, 1998.

[3] Jim Gray and Franco Putzolu. *The 5 minute rule for trading memory for disc accesses and the 10 byte rule for trading memory for CPU time*, volume 16. ACM, 1987.