**With an average page-fault service time of 8 milliseconds and a memoryaccess time of 200 nanoseconds, the effective access time in nanoseconds Is**

**effective access time = (1 − p) × (200) + p (8 milliseconds)**

**= (1 − p) × 200 + p × 8,000,000**

**= 200 + 7,999,800 × p.**

**We see, then, that the effective access time is directly proportional to the page-fault rate. If one access out of 1,000 causes a page fault, the effective access time is 8.2 microseconds. The computer will be slowed down by a factor of 40 because of demand paging!**

**Answer/Explanation**

To calculate the slowdown factor caused by demand paging, we compare the effective access time with and without demand paging.

Without demand paging, the effective access time is simply the memory access time, which is 200 nanoseconds.

With demand paging, the effective access time takes into account both the time spent on memory access (200 nanoseconds) and the average page-fault service time (8 milliseconds) weighted by the page-fault rate (1 out of 1,000 accesses).

Using the provided formula:



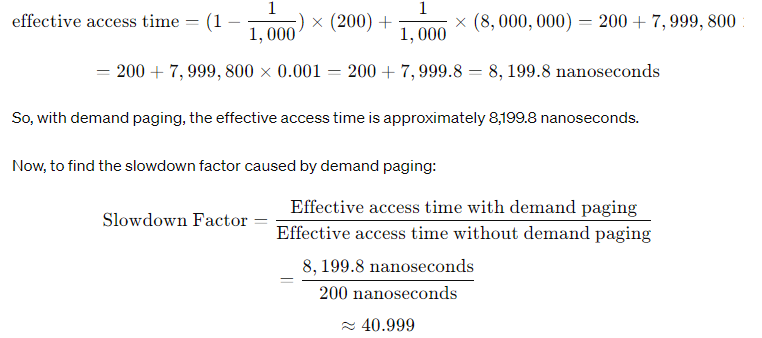
Where:

* P is the page-fault rate (1 out of 1,000 accesses).

Plugging in the values:

So, with demand paging, the effective access time is approximately 8,199.8 nanoseconds.

Now, to find the slowdown factor caused by demand paging:



Rounding it, the slowdown factor is approximately 41.

Therefore, the computer will be slowed down by a factor of approximately 41 because of demand paging.

I**f we want performance degradation to be less than 10 percent, we need**

**220 > 200 + 7,999,800 × p,**

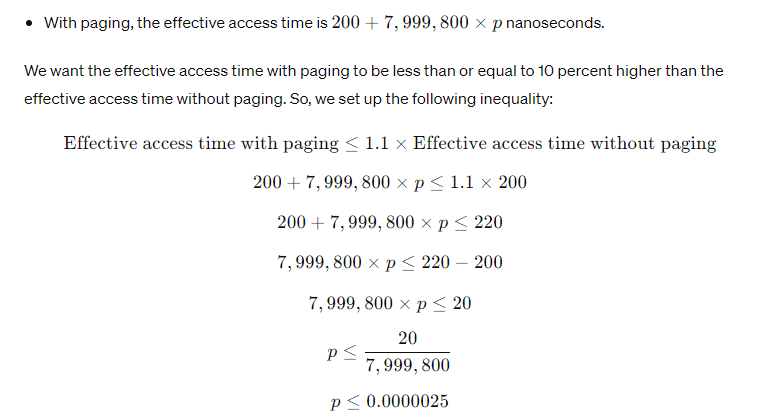
**20 > 7,999,800**

**p < 0.0000025.**

**That is, to keep the slowdown due to paging at a reasonable level, we can allow fewer than one memory access out of 399,990 to page-fault**

To ensure that the performance degradation due to paging is less than 10 percent, we need to limit the effective access time to be within 10 percent of the access time without paging.

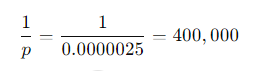
Given:



**So, to keep the slowdown due to paging at a reasonable level (less than 10 percent), we can allow fewer than one memory access out of 399,990399,990 to result in a page fault.**

In other words, for every 399,990399,990 memory accesses, we can allow at most one page fault to occur to maintain the desired performance level. This ensures that the additional time spent on handling page faults does not significantly degrade the overall system performance.

We were given the page fault rate,p=0.0000025, which represents the probability of a page fault per memory access. Therefore, to find out how many memory accesses would result in, on average, one page fault, we need to invert this probability:



This means that, on average, there will be one page fault for every 400,000 memory accesses.

So, to keep the performance degradation due to paging at an acceptable level, we can allow one page fault for every 400,000 memory accesses.