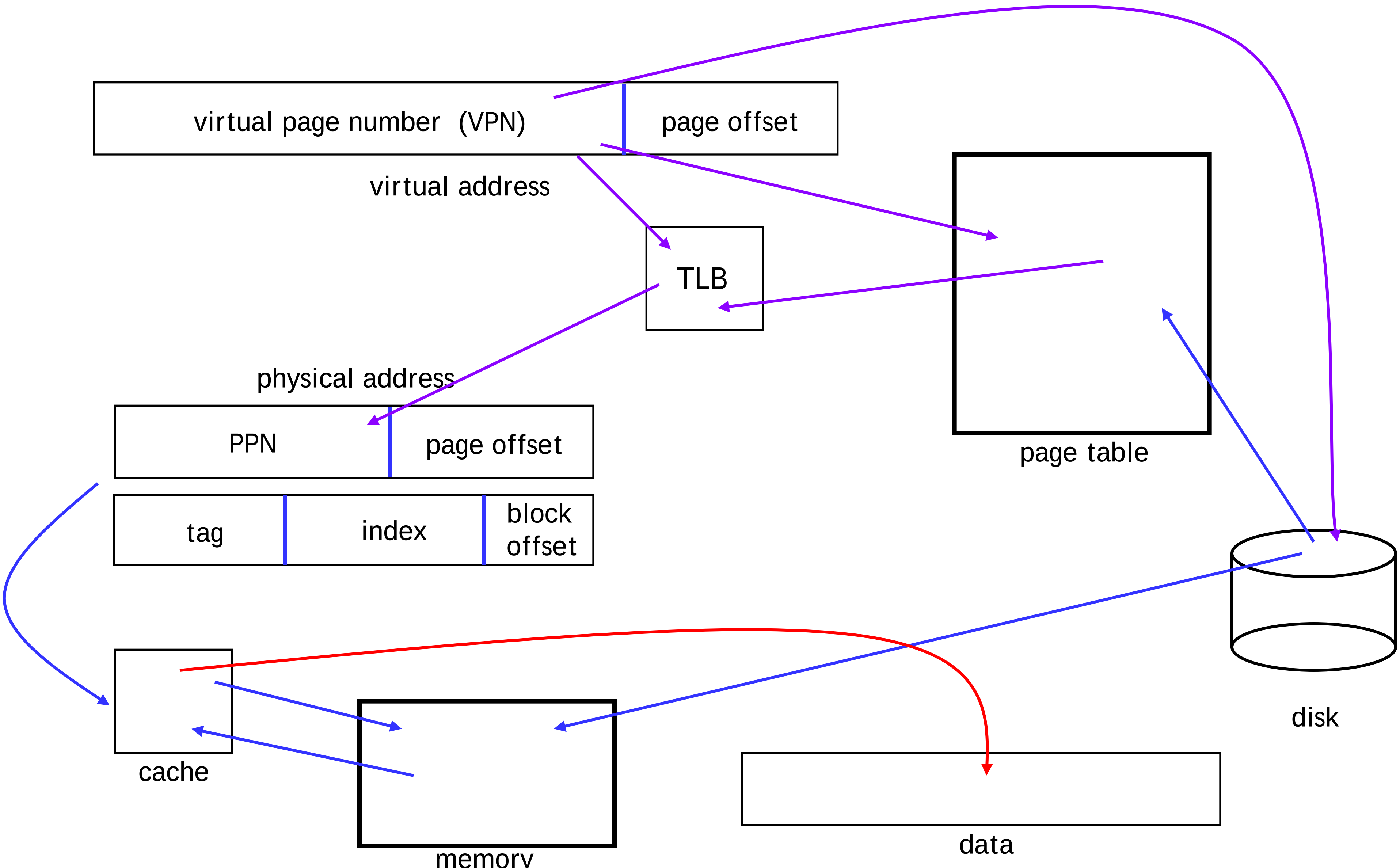
Components of the Virtual Memory System

- Arrows indicate what happens on a lw



I/O Performance

- There are two fundamental performance metrics for I/O systems:

1. Latency: the time taken for the smallest transfer (units = time)
   - This is a primary concern for programs that do many small dependent transfers

2. Bandwidth: the amount of data that can be transferred in unit time (units = bytes/time)
   - This is a primary concern for applications which transfer large amounts of data in big blocks
   - If you download large files, bandwidth will be the limiting factor

Back of the Envelope Calculation

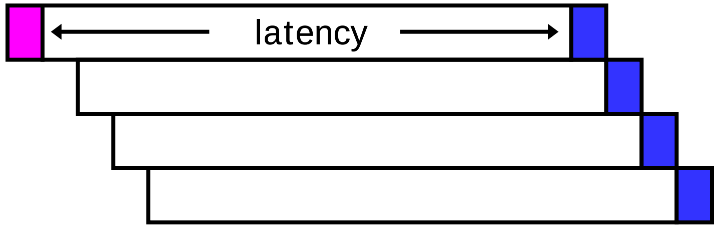
- Because data transmission can be pipelined, the total time to get data can be estimated as:

Time = latency + transfer_size / bandwidth

sec bytes / (bytes/sec)

Dominant term for small transfers

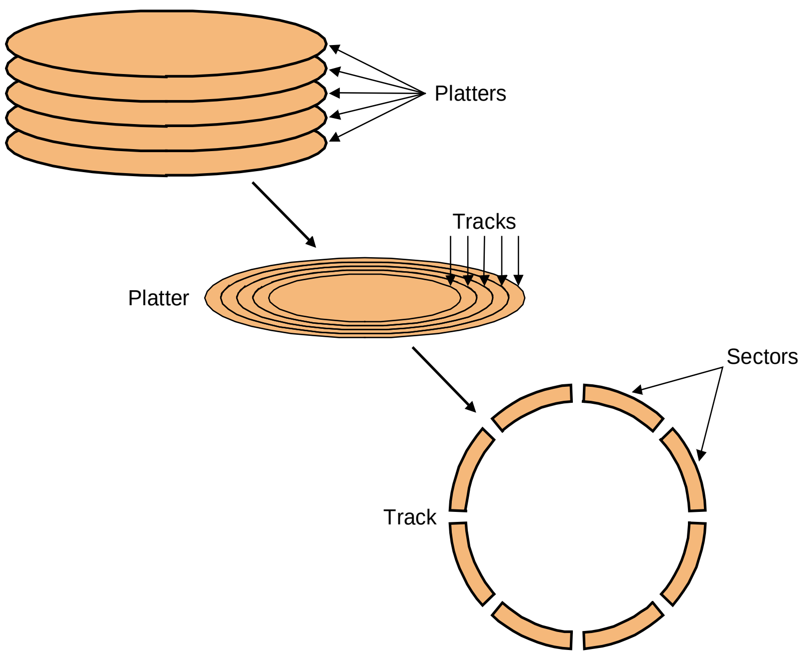
Dominant term for large transfers



Hard drives

- The textbook shows the ugly guts of a hard disk
  - Data is stored on double-sided magnetic disks called platters
  - Each platter is arranged like a record, with many concentric tracks
  - Tracks are further divided into individual sectors, which are the basic unit of data transfer
  - Each surface has a read/write head like the arm on a record player, but all the heads are connected and move together

- A 75GB IBM Deskstar has roughly:
  - 5 platters (10 surfaces),
  - 27,000 tracks per surface,
  - 512 sectors per track, and
  - 512 bytes per sector



Accessing data on a hard disk

- Factors affecting latency:
  - Seek time measures the delay for the disk head to reach the track
  - A rotational delay accounts for the time to get to the right sector

- Factors affecting bandwidth:
  - The transfer time is how long the actual data read or write takes.
  - We usually assume that the disk can read/write as fast as it can spin.
  - Thus, the transfer time is determined by the rotational speed, which also determines the rotational delay
- Manufacturers often report *average* seek times of 8-10ms
  - average the time to seek from any track to any other track

- In practice, seek times are often much better
  - if the head is already on or near the desired track, then seek time is much smaller (2-3ms) — locality is important!

Estimating rotational delay

- Once the head is in place, we need to wait until the right sector is underneath the head
  - This may require as little as no time (reading consecutive sectors) or as much as a full rotation (just missed it)
  - On average, for random reads/writes, we can assume that the disk spins halfway on average

- Rotational delay depends on how fast the disk platters spin:

  Average rotational delay = 0.5 rotations / rotations per minute

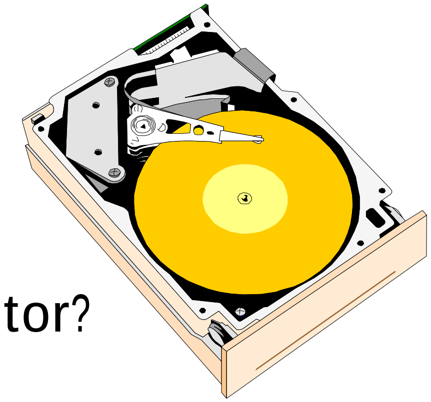
  - For example, a 5400 RPM disk has an average rotational delay of:

    0.5 rotations / (5400 rotations/minute) = 5.55ms

- The average latency is the sum of the average seek time and the average rotational delay

Estimating transfer time

- Assume a disk has the following specifications
  - An average seek time of 11ms
  - A 5400 RPM rotational speed
  - A 10MB/s average transfer rate



- How long does it take to read a random 1,024 byte sector?
  - The average rotational delay is 5.55ms
  - Thus, the average latency is 11 + 5.55 = 16.55ms

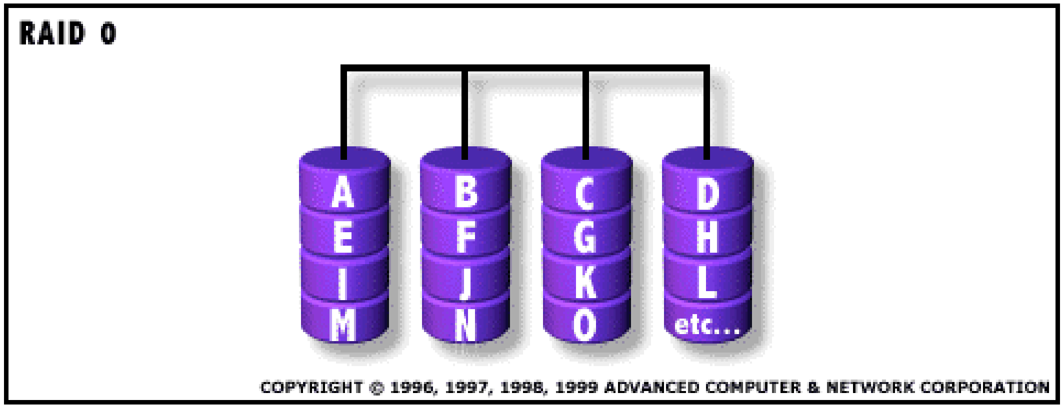
- The transfer time will be about (1024 bytes / 10 MB/s) = 0.1ms

- It thus takes 16.55ms + 0.1ms = 16.7ms to read a random sector
  - That's 16,700,000 cycles for a 1GHz processor!

- One possible measure of bandwidth would be the number of consecutive sectors that can be read in one second
  - each additional sector takes 0.1ms to read

Parallel I/O

- Many hardware systems use parallelism for increased speed
  - Pipelined processors include extra hardware so they can execute multiple instructions simultaneously
  - Dividing memory into banks lets us access several words at once
- A redundant array of inexpensive disks or RAID system allows access to several hard drives at once, for increased bandwidth
  - The picture below shows a single data file with fifteen sectors denoted A-O, which are “striped” across four disks
  - This is reminiscent of interleaved main memories from last week



Inferring the cache structure

- Consider the following program

```
char a[LENGTH]; int sum = 0;
for(int i = 0; i < 10000; ++i)
  for(int j = 0; j < LENGTH; j += STEP)
    sum += a[j];
```

- Key idea: compute the *average* time it takes to execute **sum += a[j]**

average time (in ns)

	STEP							
LENGTH	1	2	4	8	16	32	64	128
8	7	7	7	7	6	7	7	7
16	8	7	7	6	6	7	7	7
32	7	8	7	7	7	7	6	7
64	16	21	45	45	7	8	8	9
128	14	25	47	45	45	7	7	7
256	17	25	43	45	45	46	8	7

- What is the cache size (data)? What is the block size? What is the associativity?