PERSISTENCE: DISK SCHEDULING

Shivaram Venkataraman
CS 537, Spring 2023
Project 4 grades out. Regrades?

Project 5 – due soon?

Midterm 2 – April 4th, lots of details on Piazza
AGENDA / LEARNING OUTCOMES

How do you calculate sequential and random tput of a disk?

What algorithms are used to schedule I/O requests?
RECAP
while (STATUS == BUSY)
;
// spin
Write data to DATA register
Write command to COMMAND register
while (STATUS == BUSY)
;
// spin
Motor connected to spindle spins platters

Rate of rotation: RPM

10,000 RPM → single rotation is 6 ms
Heads on a moving arm can read from each surface.
SEEK, ROTATE, TRANSFER

Seek cost: Function of cylinder distance
   Not purely linear cost
   Must accelerate, coast, decelerate, settle
   Settling alone can take 0.5 - 2 ms

Entire seeks often takes 4 - 10 ms
Average seek = 1/3 of max seek

Total time = seek + rotation + transfer time

Depends on rotations per minute (RPM)
   7200 RPM is common, 15000 RPM is high end

Average rotation: Half of time for 1 rotation

Pretty fast: depends on RPM and sector density.
   100+ MB/s is typical for maximum transfer rate
What is the time for 4KB random read with Cheetah?

<table>
<thead>
<tr>
<th></th>
<th>Cheetah 15K.5</th>
<th>Barracuda</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacity</td>
<td>300 GB</td>
<td>1 TB</td>
</tr>
<tr>
<td>RPM</td>
<td>15,000</td>
<td>7,200</td>
</tr>
<tr>
<td>Average Seek</td>
<td>4 ms</td>
<td>9 ms</td>
</tr>
<tr>
<td>Max Transfer</td>
<td>125 MB/s</td>
<td>105 MB/s</td>
</tr>
<tr>
<td>Platters</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Cache</td>
<td>16 MB</td>
<td>16/32 MB</td>
</tr>
<tr>
<td>Connects via</td>
<td>SCSI</td>
<td>SATA</td>
</tr>
</tbody>
</table>
What is the time for 4KB random read with Barracuda?

<table>
<thead>
<tr>
<th></th>
<th>Cheetah 15K.5</th>
<th>Barracuda</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacity</td>
<td>300 GB</td>
<td>1 TB</td>
</tr>
<tr>
<td>RPM</td>
<td>15,000</td>
<td>7,200</td>
</tr>
<tr>
<td>Average Seek</td>
<td>4 ms</td>
<td>9 ms</td>
</tr>
<tr>
<td>Max Transfer</td>
<td>125 MB/s</td>
<td>105 MB/s</td>
</tr>
<tr>
<td>Platters</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Cache</td>
<td>16 MB</td>
<td>16/32 MB</td>
</tr>
<tr>
<td>Connects via</td>
<td>SCSI</td>
<td>SATA</td>
</tr>
</tbody>
</table>
WORKLOAD PERFORMANCE

So…
- seeks are slow
- rotations are slow
- transfers are fast

How does the kind of workload affect performance?
Sequential: access sectors in order
Random: access sectors arbitrarily
## Disk Spec

<table>
<thead>
<tr>
<th></th>
<th>Cheetah</th>
<th>Barracuda</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacity</td>
<td>300 GB</td>
<td>1 TB</td>
</tr>
<tr>
<td>RPM</td>
<td>15,000</td>
<td>7,200</td>
</tr>
<tr>
<td>Avg Seek</td>
<td>4 ms</td>
<td>9 ms</td>
</tr>
<tr>
<td>Max Transfer</td>
<td>125 MB/s</td>
<td>105 MB/s</td>
</tr>
<tr>
<td>Platters</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Cache</td>
<td>16 MB</td>
<td>32 MB</td>
</tr>
</tbody>
</table>

Sequential read 100MB: what is throughput for each?
I/O SCHEDULERS
I/O SCHEDULERS

Given a stream of I/O requests, in what order should they be served?

Much different than CPU scheduling

Position of disk head relative to request position matters more than length of job
FCFS (FIRST-COME-FIRST-SERVE)

Assume seek+rotate = 10 ms for random request

How long (roughly) does the below workload take? Requests are given in sector numbers

300001, 700001, 300002, 700002, 300003, 700003

300001, 300002, 300003, 700001, 700002, 700003
SSTF (SHORTEST SEEK TIME FIRST)

Strategy always choose request that requires least seek time
(approximate total time with seek time)

Greedy algorithm (just looks for best NEXT decision)

How to implement in OS?

Disadvantages?
SCAN or Elevator Algorithm:

- Sweep back and forth, from one end of disk other, serving requests as pass that cylinder
- Sorts by cylinder number; ignores rotation delays

C-SCAN (circular scan): Only sweep in one direction

Pros/Cons?
SPTF (SHORTEST POSITIONING TIME FIRST)

Rotates this way

SATF (SHORTEST ACCESS TIME FIRST)
Disk accesses: 32, 12, 33, 3, 13, 4
Rotation Time = 2ms (non-adjacent reads)
Seek Time (for adjacent track) = 2ms.

What is the time taken when using (FCFS) scheduling?
Disk accesses: 32, 12, 33, 3, 13, 4
Rotation Time = 2ms (non-adjacent reads)
Seek Time (for adjacent track) = 2ms.

Order in which requests will be serviced for Shortest Seek Time First (SSTF)?

Time Taken
Schedulers

Where should the scheduler go?
void reader(int fd) {
    char buf[1024];
    int rv;
    while((rv = read(fd, buf)) != 0) {
        assert(rv);
        // takes short time, e.g., 1ms
        process(buf, rv);
    }
}

Assume 2 processes each calling read() with C-SCAN
Work conserving schedulers always try to do work if there's work to be done.

Sometimes, it's better to wait instead if system anticipates another request will arrive.

Possible improvements from I/O Merging.
SUMMARY

Disks: Specific geometry with platters, spindle, tracks, sector

I/O Time: rotation_time + seek_time + transfer_time
Sequential throughput vs. random throughput

Scheduling approaches: SSTF, SCAN, C-SCAN
Benefits of violating work conservation
Next class: How to achieve resilience against disk errors