Research on Disks and Disk Scheduling

Brian Railing

Monday, November 3rd 2003

15-410 Fall 2003

Original lecture given by Steve Muckle on Monday, March 31st 2003
Additional Slides Taken from Eno Thereska's July Systems Talk
Also from Terrence Wong's Midsemester Thesis Presentation

Outline

- Freeblock Scheduling
- Timing Accurate Storage Emulation (TASE)
- Self-*

Freeblock Scheduling

- Research going on right here at CMU
- Something I was involved in this past summer
- Who would like some free bandwidth while their disk is busy?

Freeblock Scheduling

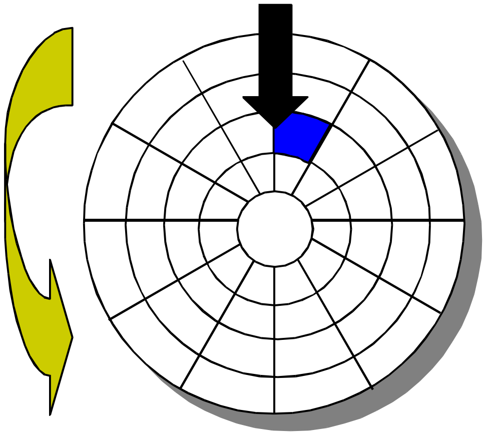
- Interface:
  - fb_read(logical numbers, …)
  - callback_fn(…)

- Extracting Bandwidth
  - Send requests to the disk in between normal requests without effecting the normal requests

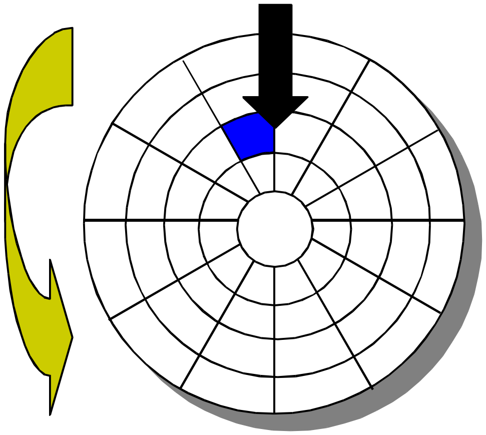
Freeblock Scheduling

- As in SPTF scheduling, we must know the EXACT state of the disk
- We need to be able to predict how much rotational latency we have to work with
- Enemies of freeblock scheduling:
  disk prefetching
  internal disk cache hits
  unexpected disk activity (recalibration, etc)
  disk-reordered requests

About to read blue sector

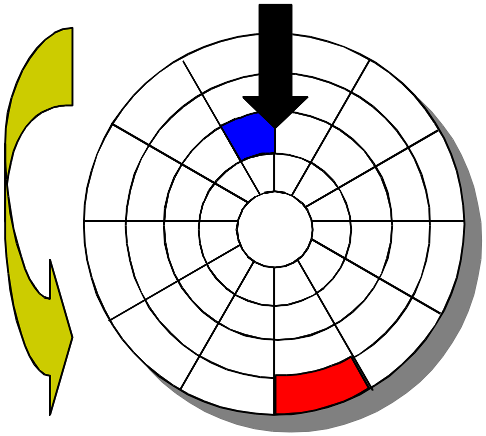


After reading blue sector

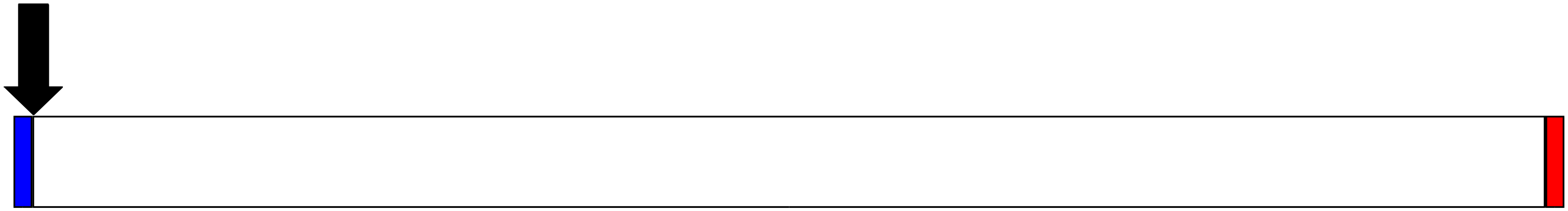


After BLUE read

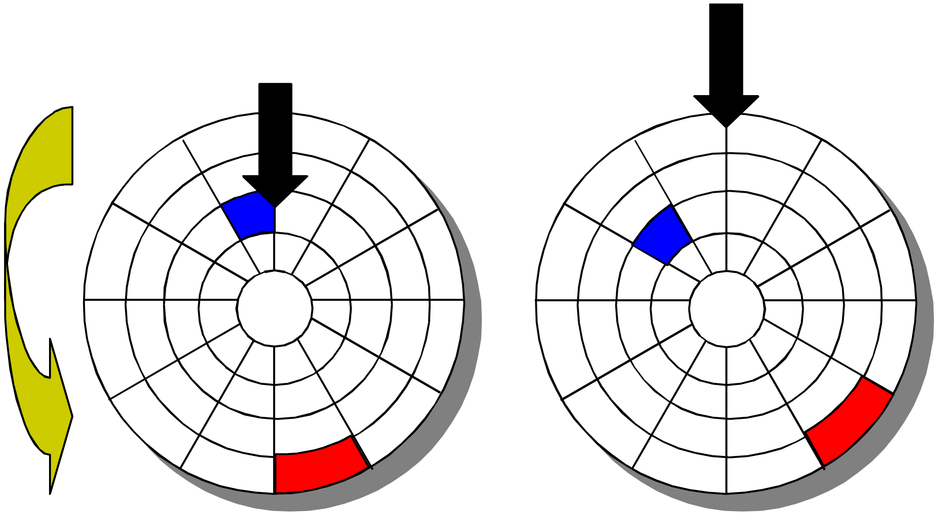
Red request scheduled next



After BLUE read

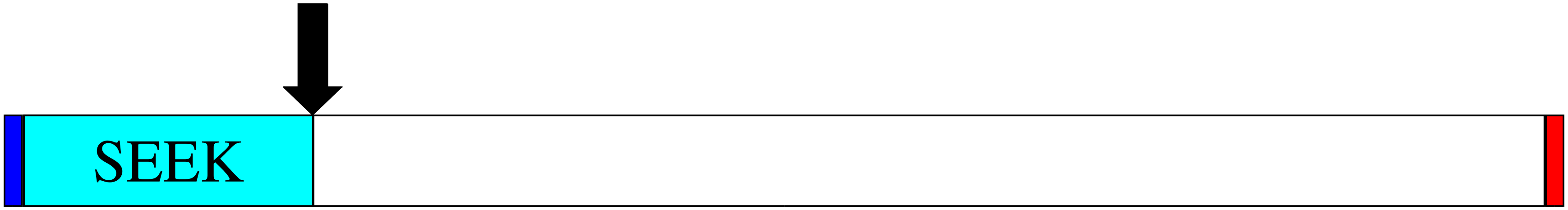


Seek to Red’s track

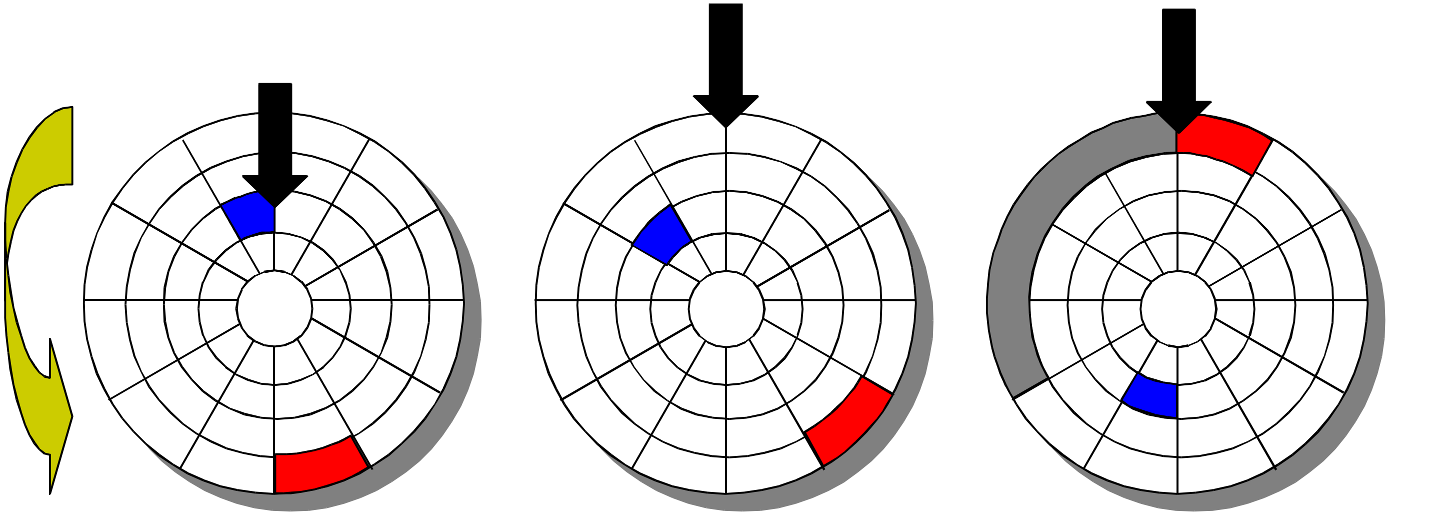


After BLUE read

Seek for RED



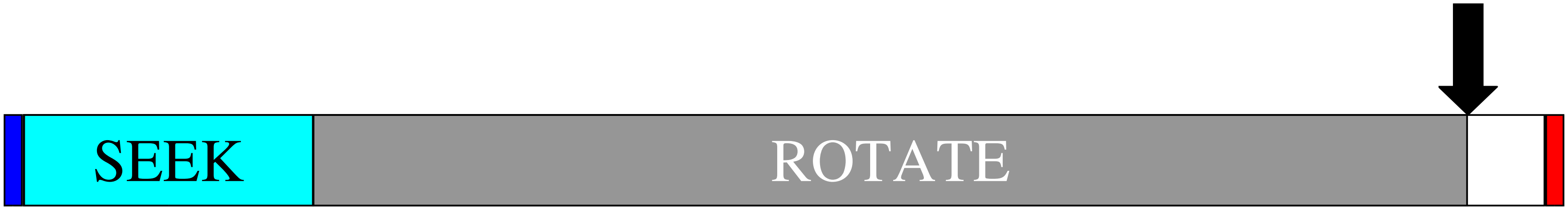
Wait for Red sector to reach head



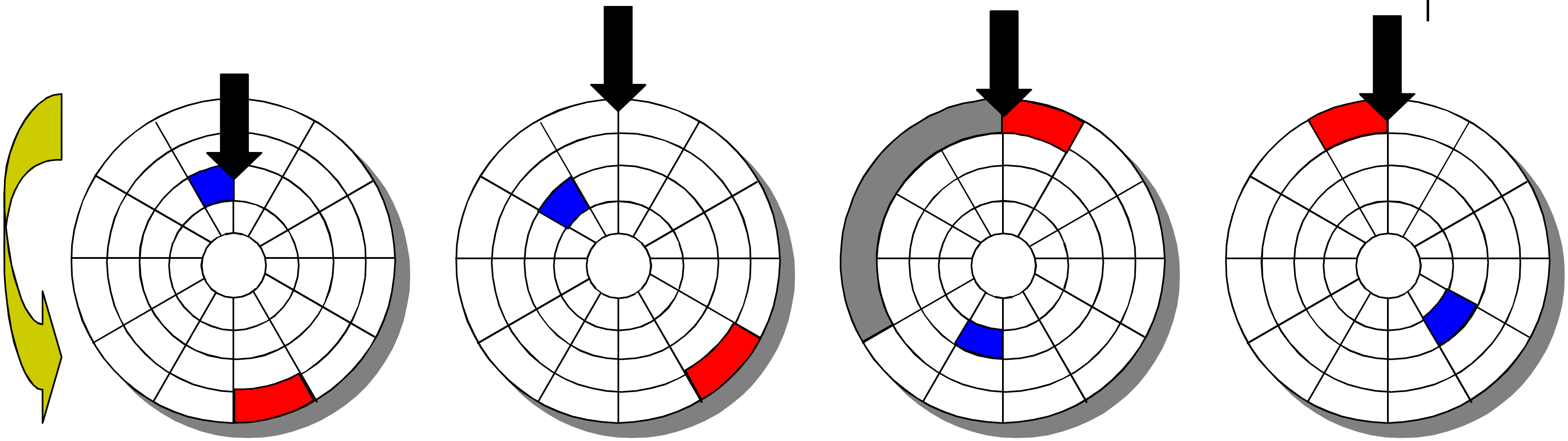
After BLUE read

Seek for RED

Rotational latency



Read Red sector

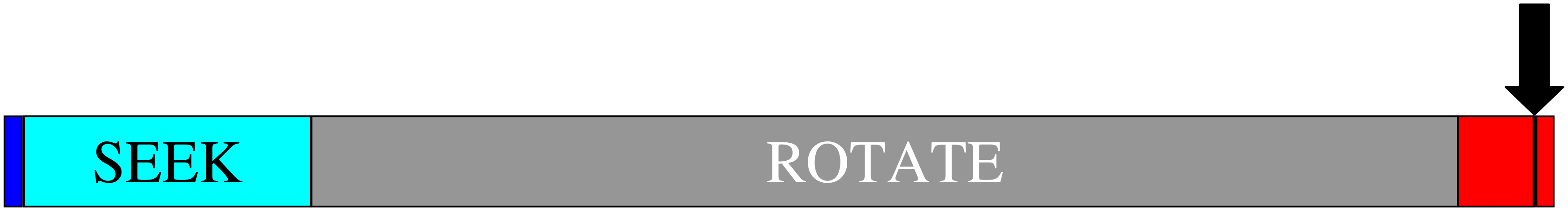


After BLUE read

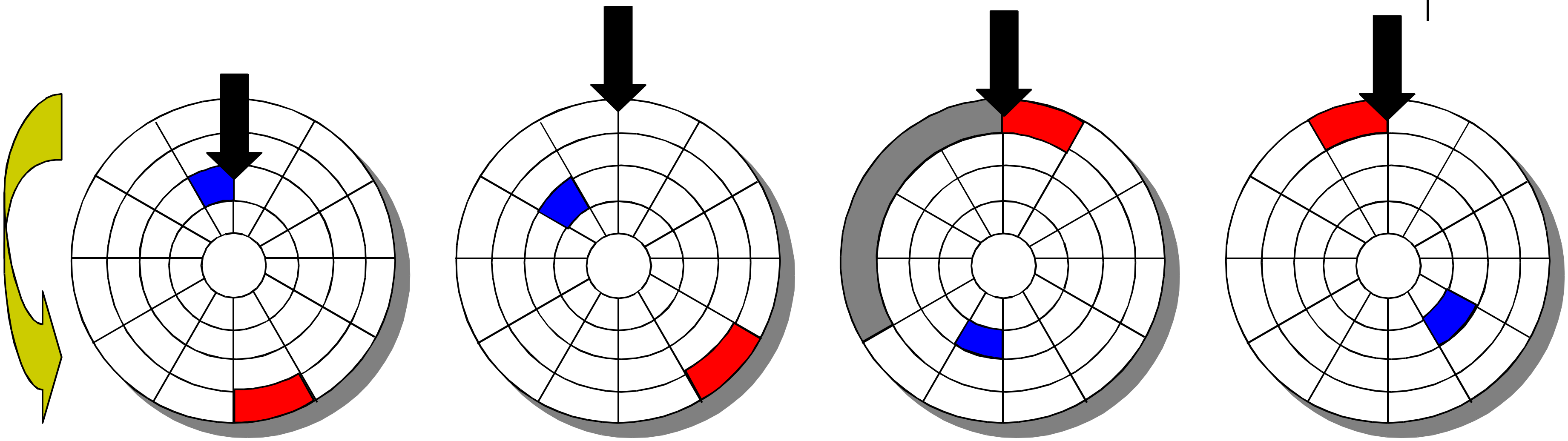
Seek for RED

Rotational latency

After RED read



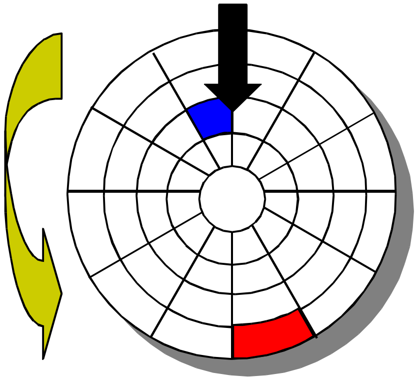
Traditional components



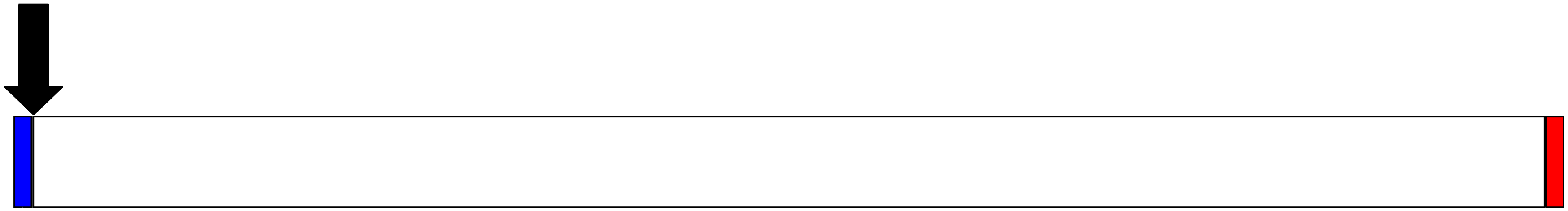
After BLUE read Seek for RED Rotational latency After RED read

- Note: Rot. Latency is an artifact of rotation
    - Seeks are needed to keeps disk head on tracks

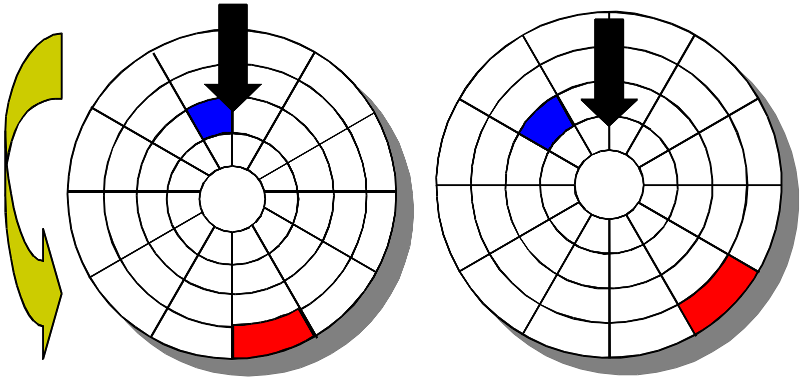
Initial setup again



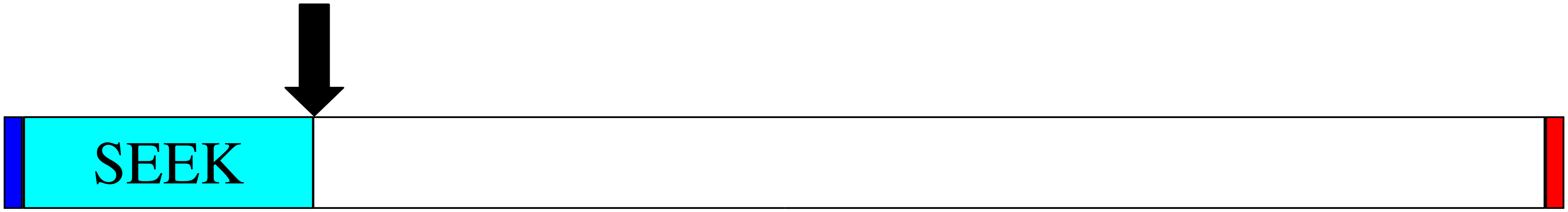
After BLUE read



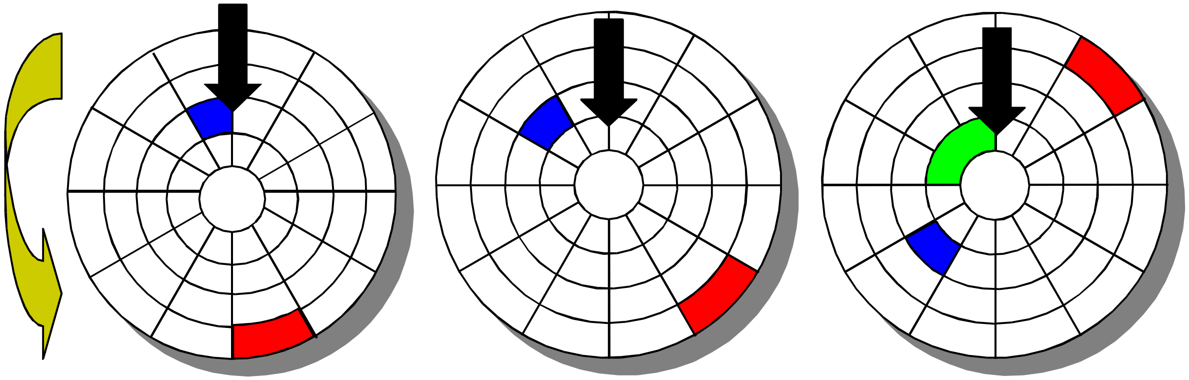
Seek to Third track



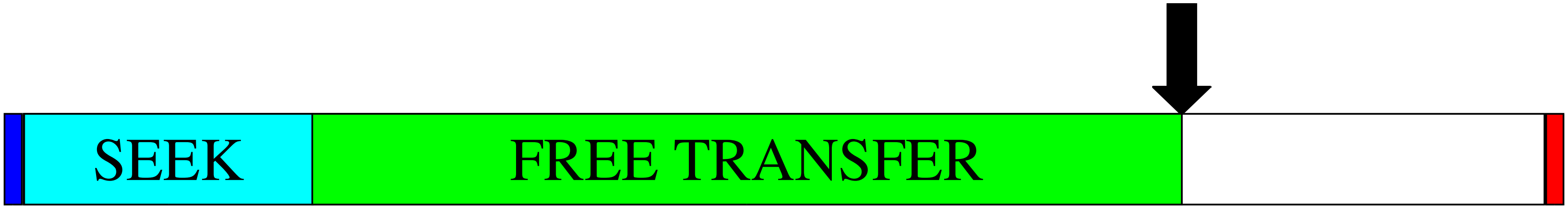
After BLUE read Seek to Third



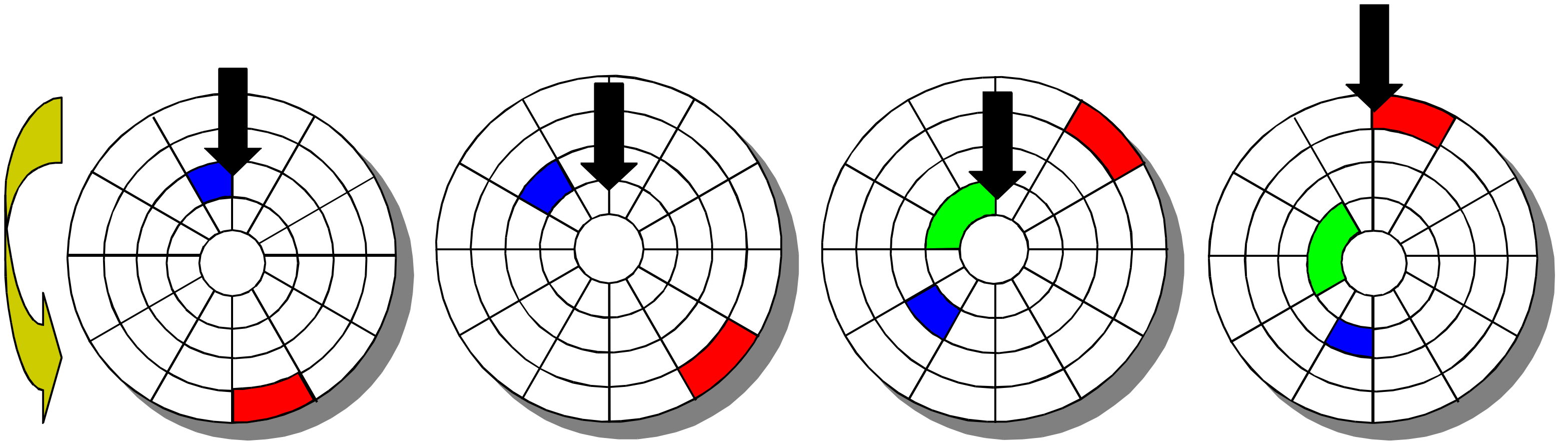
Free transfer



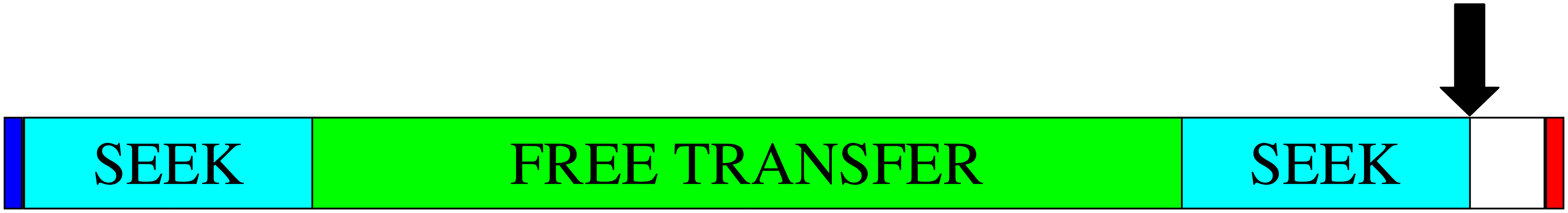
After BLUE read Seek to Third Free transfer



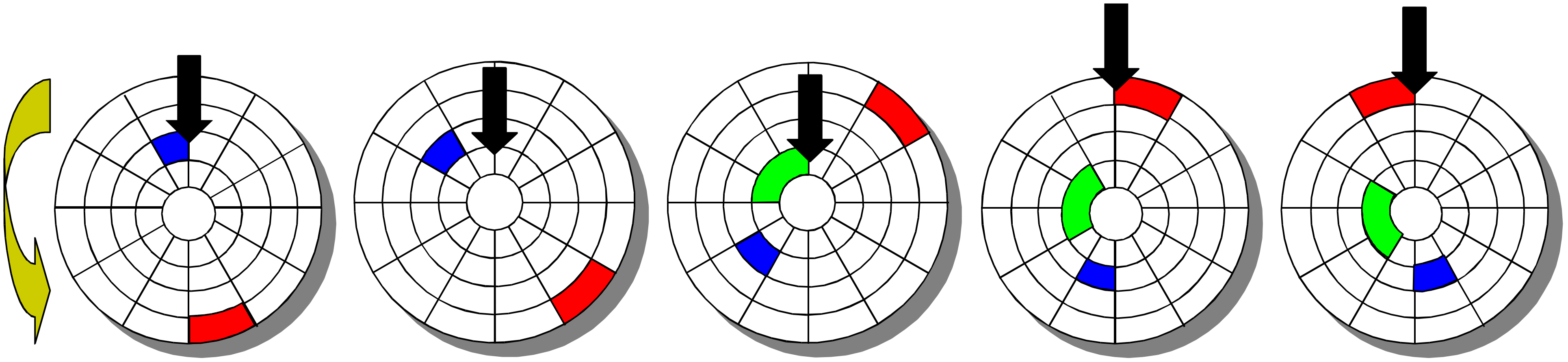
Seek to Red’s track



After BLUE read Seek to Third Free transfer Seek to RED



Read Red sector



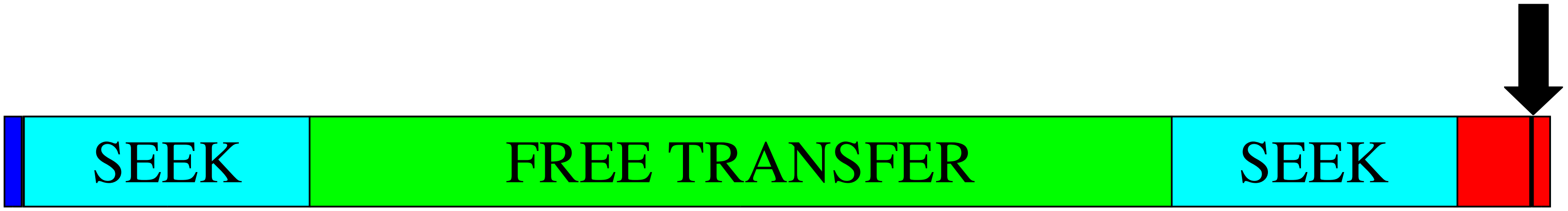
After BLUE read

Seek to Third

Free transfer

Seek to RED

After RED read

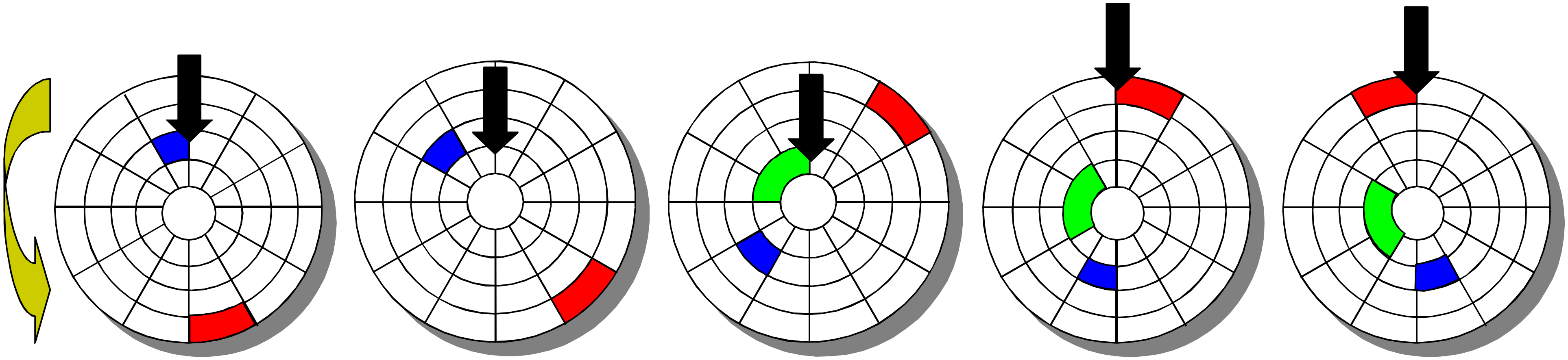


SEEK

FREE TRANSFER

SEEK

Resulting components



After BLUE read Seek to Third Free transfer Seek to RED After RED read

- Interesting, but can apps use free bw?

Freeblock Scheduling

- Results include 3.1MB/sec of free bandwidth
- This free bandwidth is best suited to applications with loose time constraints
- Some sample applications:
  - backup applications
  - disk array scrubbing
  - cache cleaning (perhaps…)

TASE

- Research I'm currently involved with

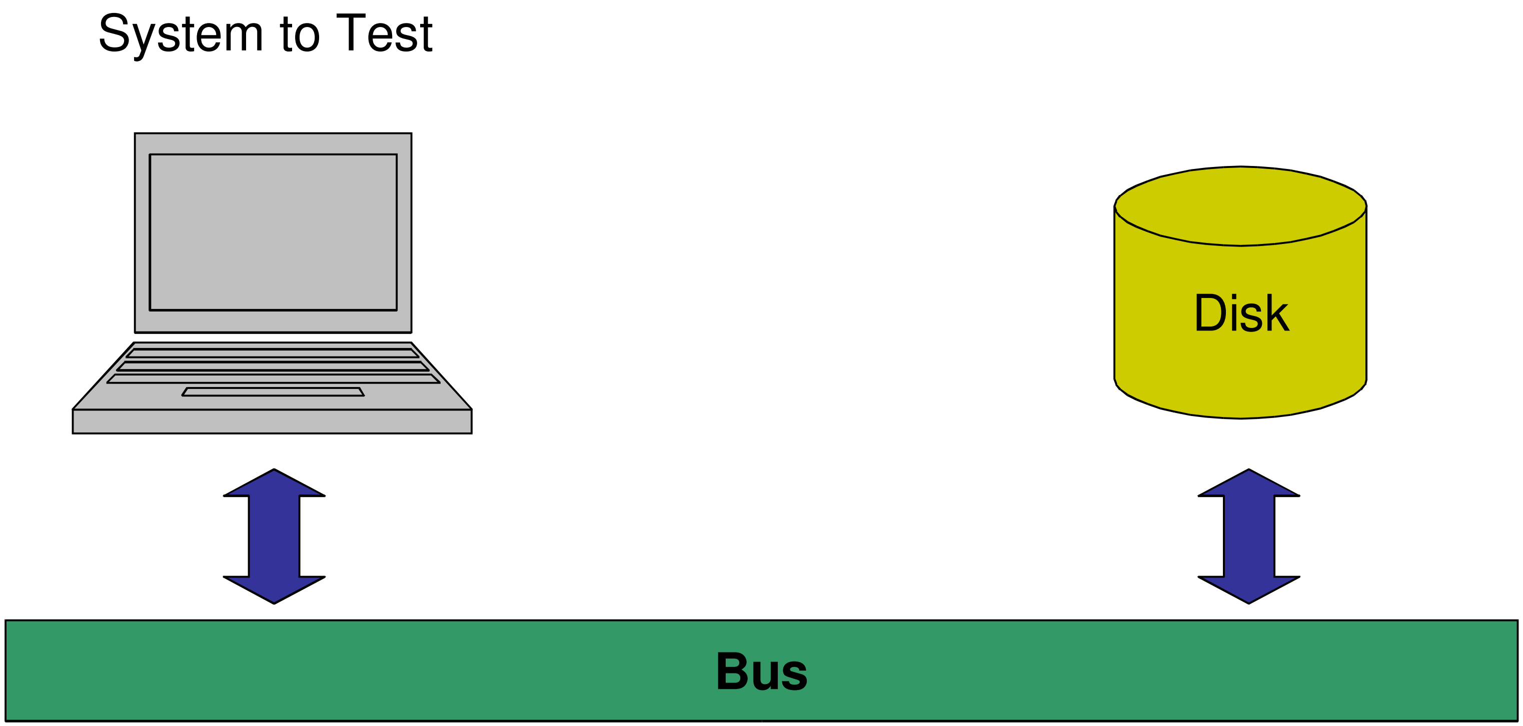
- Timing accurate
  - Can get performance measurements

- Evaluate hypothetical storage devices
  - Without building a prototype
  - In real systems

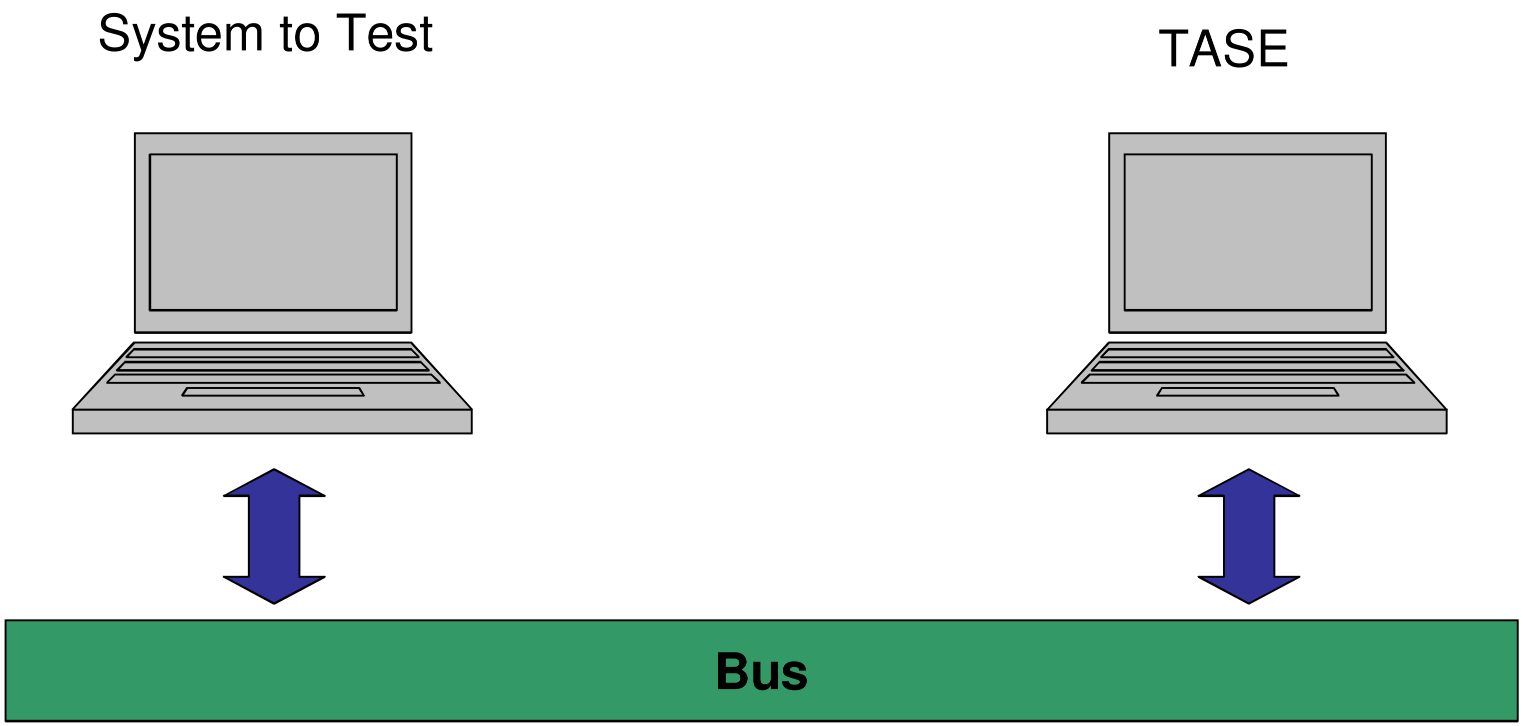
TASE

- Storage Evaluation Techniques
  - Hand calculations
  - Simulation
  - Emulation
  - Prototypes
  - Real System

TASE



TASE



System to Test

TASE

Bus

TASE

“If it walks like a duck and talks like a duck, it must be a duck.”

- Emulated device needs to “be” a disk
  - Respond over bus to system being tested
  - Behave like a disk by storing requests

TASE

- Everything needs to be in physical memory
  - This limits what we can test with the device

- Possible Solutions
  - Use multiple machines as emulators
  - Compress data
  - Find data that doesn’t need to be stored

TASE

- Two expectations of disks
  - Data is accessible
  - It is returned correctly

- Do we have to meet these expectations?

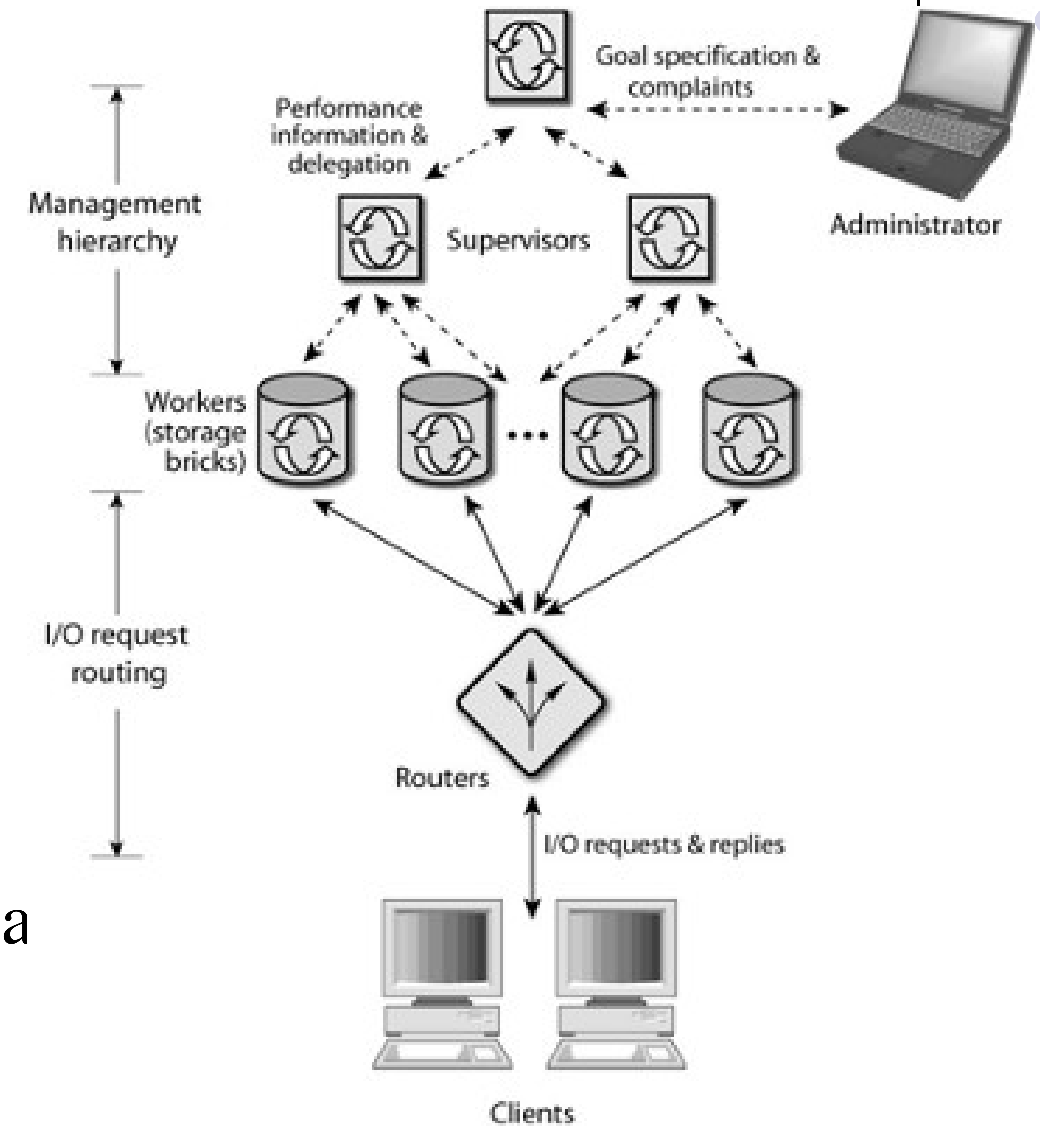
Self-*

- Storage Management
  - Currently: 1 admin per 1 - 10TB
  - Goal is to increase to 1 admin per 1PB

- What is necessary to allow this increase?
  - Could wait for hardware improvements
  - Or we could do research

Self-*

- Self-*
  1. Petabyte scale
  2. Self-organizing
  3. Self-managing
  4. Self-tuning
  5. Self-configuring
  6. Self-repairing
  7. Commodity hardwa



Related Reading

- Freeblock Scheduling http://www.pdl.cmu.edu/Freeblock/index.html
- TASE http://www.pdl.cmu.edu/PDL-FTP/Storage/timing_abs.html
- Self-* http://www.pdl.cmu.edu/SelfStar/index.html

Conclusions

- Much research into improving disk access

- This is just a small part of current research

- Part of idea behind doing the book report