DATA STORAGE TECHNOLOGIES & NETWORKS

(CS C446, CS F446 & IS C446)

LECTURE 18– STORAGE

Reducing Energy Consumption in Disks

Component	Manufacturer & Model	Power (watts)	Percent of Total
Display	Compaq monochrome lite25c	3.5	68%
Disk Drive (105 Mbytes)	Maxtor MXL-105 III	1.0	20%
CPU	3.3V Intel486	0.6	12%
Memory (16 Mbytes)	Micron MT4C4M4A1/B1	0.024	0.5%

Kester Li., et. al., A Quantitative analysis of disk power management in portable computers

Reducing Energy Consumption in Disks

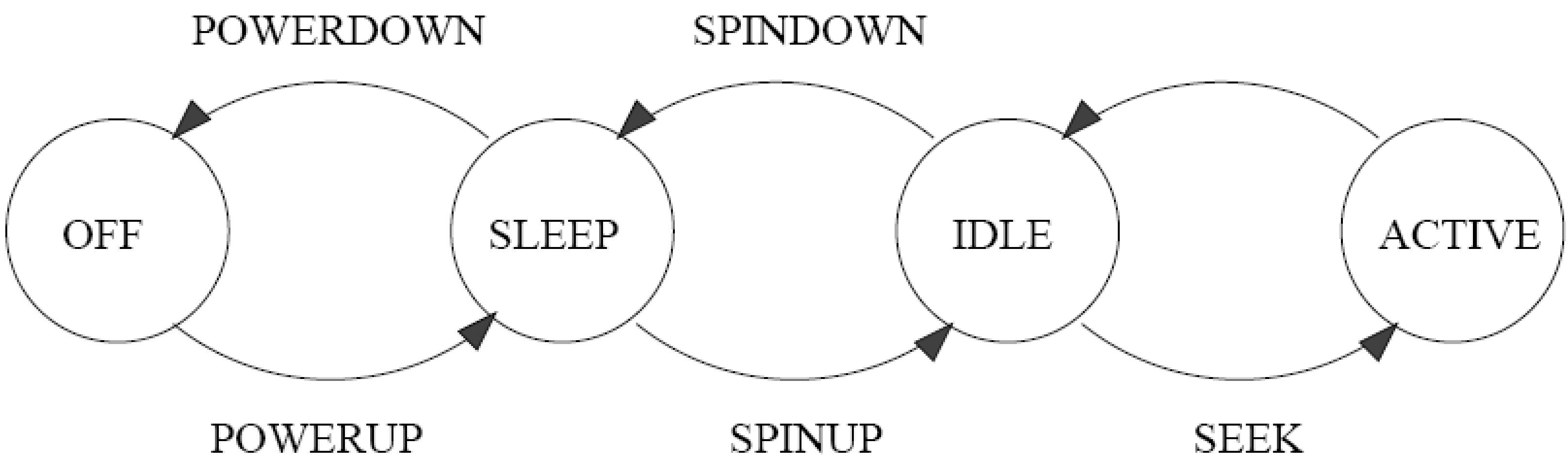

Figure 1: **Disk State Transition Diagram.**

Mode	Power (watts)
OFF	0.0
SLEEP	0.025
IDLE	1.0
ACTIVE	1.95

Transition	Time (seconds)	Power (watts)
POWERUP	0.5	0.025
SPINUP	2.0	3.0
SEEK	0.009	1.95
SPINDOWN	1.0	0.025
POWERDOWN	0.5	N/A

- Results
  - 2 seconds spin down delay saves 90% energy
  - Takes additional time – reduces the performance slightly
  - Can overcome by disk caches

Efficient Scheduling - Issues

- Major power management considerations
  - For processors supporting DVS/DFS, the memory, hard disk and networking devices typically support a small number of discrete states – with varying power consumption and access time
  - Move a device into one of its low power state during the period of idle time. Challenges:
    - Need to identify the state that will maximize energy saving for a given level of acceptable performance degradation

Increasing Disk Burstiness for Energy Efficiency by Athanasios E. Papathanasiou and Michael L. Scott

Efficient Scheduling - Issues

- Challenges Continues
  - The optimal power saving depends not only on the device characteristics but also on the length of idle time which is not known apriori. Common policies:
    - After certain period of device inactivity – Simple to implement
    - monitor the utilization pattern of the underlying device, keep track of the length and frequency of idle periods, and attempt to select a power mode based on prediction of the duration of the upcoming idle period

Efficient Scheduling - Issues

- Reality
  - If the time between device accesses is too small to justify moving to a low power mode, then the most efficient reactive policy will keep the device constantly active, and no energy will be saved.
- How to improve the performance
  - *Change* the access pattern.
  - OS should deliberately attempt to create *bursty* access patterns for devices with low power modes
    - Scheduling, Memory management and Buffering is under the control of OS
    - Delay asynchronous read and write requests, prefetch aggressively to reduce synchronous reads from disk

Advanced Configuration and Power Interface Specification (ACPI)

- Modern hard disks for mobile systems support 4 different power states:
  - Active, Idle, Standby, and Sleep.
  - Idle state [disk still spins, but the electronics may be partially unpowered, and the heads may be parked or unloaded]
  - Standby state [disk spuns down]
  - Sleep state [powers off all remaining electronics. hard reset is required to return to higher states]
  - Individual devices may support additional states
    - The IBM TravelStar has three different Idle sub-states

Efficient Scheduling - Issues

- Idle to Active
  - Minimum time and energy requirement
- Sleep to any state
  - High time and energy requirement
- Standby to Active
  - Lies in the middle. 1 to 3 seconds and 1.5X to 2X energy requirement.

Disk	M-1994	IBM-2000	T-2001	IBM-micro
Capacity	105MB	6-18GB	2GB	340MB-1GB
Active	1.95W	2.1W	1.3W	0.73W
Idle	1.0W	1.85W	0.7W	0.5W
Active Idle	NA	0.85W	NA	NA
Low Power Idle	NA	0.65W	0.5W	0.22W
Standby	0.025W	0.25W	0.23W	0.066W
Spin up	3.0W	3.33W	3.0W	0.66W
Spin down	0.025W	NA	NA	NA
Spin up time	2.0s	1.8s	1.2s	0.5s
Spin down time	1.0s	NA	NA	NA
Breakeven time	6.2s	6.2s	7.6–13.3s	0.7–2s

- M-1994 – MAXTOR MXL-105 III (1994)
- IBM-200 – IBM Travelstar (2000)
- T-2001 – Toshiba MK5002 MPL (2001)
- IBM-micro – IBM Microdrive DSCM.

Efficient Scheduling

- To improve performance
  - *maximize the length of idle phases* by prefetching aggressively and by postponing write requests
  - *minimize wait time* in the Idle state during idle phases long enough to justify spin down
  - *avoid unnecessary spin downs* during short idle phases
  - *minimize application perceived delays* through disk pre-activation