Systems Support for Preemptive Disk Scheduling

Tae-Young Choe

choety@kumoh.ac.kr

School of Computer Engineering, KIT

References

- Sitaram Lyer and Peter Druschel, “Anticipatory scheduling: A disk scheduling framework to overcome deceptive idleness in synchronous I/O,” 18th ACM Symposium on Operating Systems Principle (SOSP 2001).

Introduction

- Motivations
  - For performance: multiple outstanding requests.
  - Each process: tends to issue contiguous disk requests.
  - In real: interspersing successive requests between short computations.
- Deceptive idleness
  - a condition where the scheduler incorrectly assumes that the last request issuing process has no further requests,
  - becomes forced to switch to a request from another process
- Pairness degrades performance: interleave IOs of multiple processes too frequently.

Anticipatory disk scheduling framework

- Before choosing a request for service,
  - it sometimes
  - introduces a short, controlled delay periods,
  - waiting for additional requests from the process.
- An application of the non-work-conserving scheduling discipline.
- Implementation
  - as a kernel module in the FreeBSD operating system

Examples of deceptive idleness

- For consecutive IO requests: throughput of 21 MB/s (service time of 3ms for every 64KB block).
- Seek reducing schedulers
  - Given two processes issuing IOs synchronously
  - throughput of 5 MB/s due to additional 9ms of ave. seek time
- Proportional-share schedulers
  - Intended proportion 1:2 cannot be maintained if the number of outstaning requests is 1 $\Rightarrow$ proportion becomes 1:1

Prefetching

- Issuing immediately forthcoming request before the current one completes
- Two types
  - Application-driven prefetch: issue asynchronous IO such aio_read
    - Applications do not exactly predict itself.
  - Kernel-driven prefetch
    - Also, kernel cannot predict exactly.

Anticipatory scheduling

- Three necessary conditions for deceptive idleness
  1. multiple disk-intensive applications concurrently issuing synchronous disk requests
  2. the intrinsic non-preemptible nature of disk requests
  3. a work-conserving disk scheduler
- Anticipatory scheduling eliminates condition 3
- The *framework* potentially waits briefly for additional requests to arrive
- Key point: whether and how long to wait?

Experimental evaluation

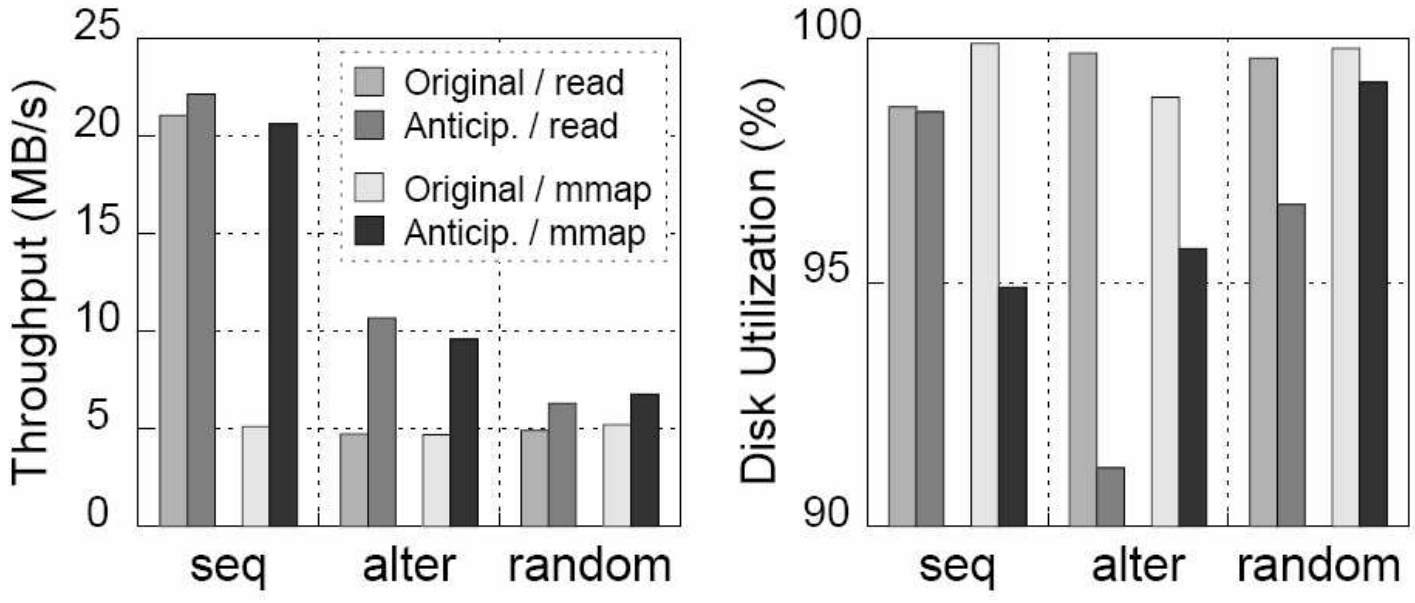
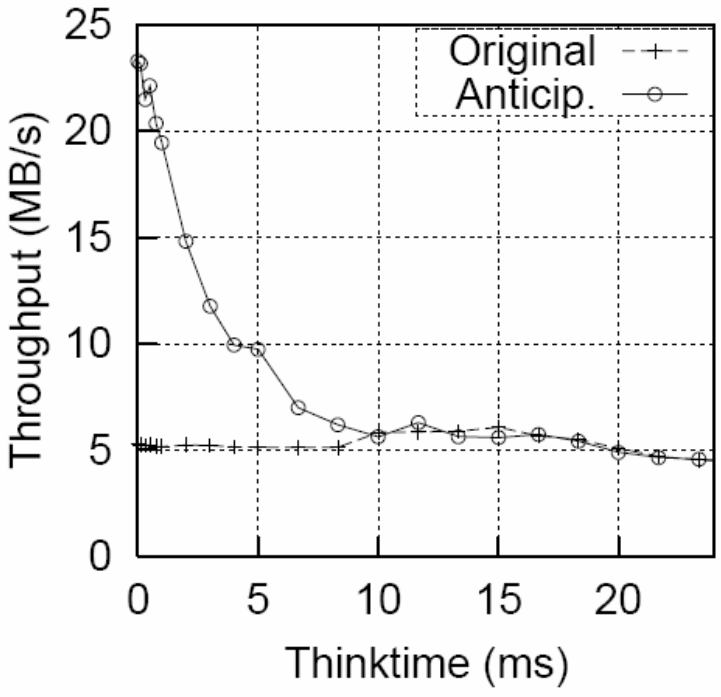


Figure 4: Impact of anticipatory scheduling on disk throughput and utilization, using sequential, alternate-block and random access workloads, and read versus mmap based access.

Experimental evaluation (Cont.)



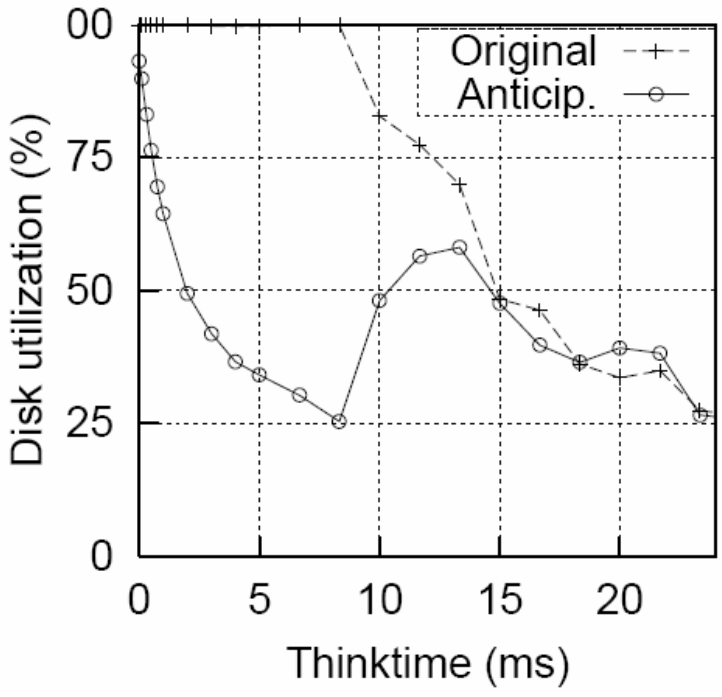


Figure 5: Increasing thinktimes for both processes

Experimental evaluation (Cont.)

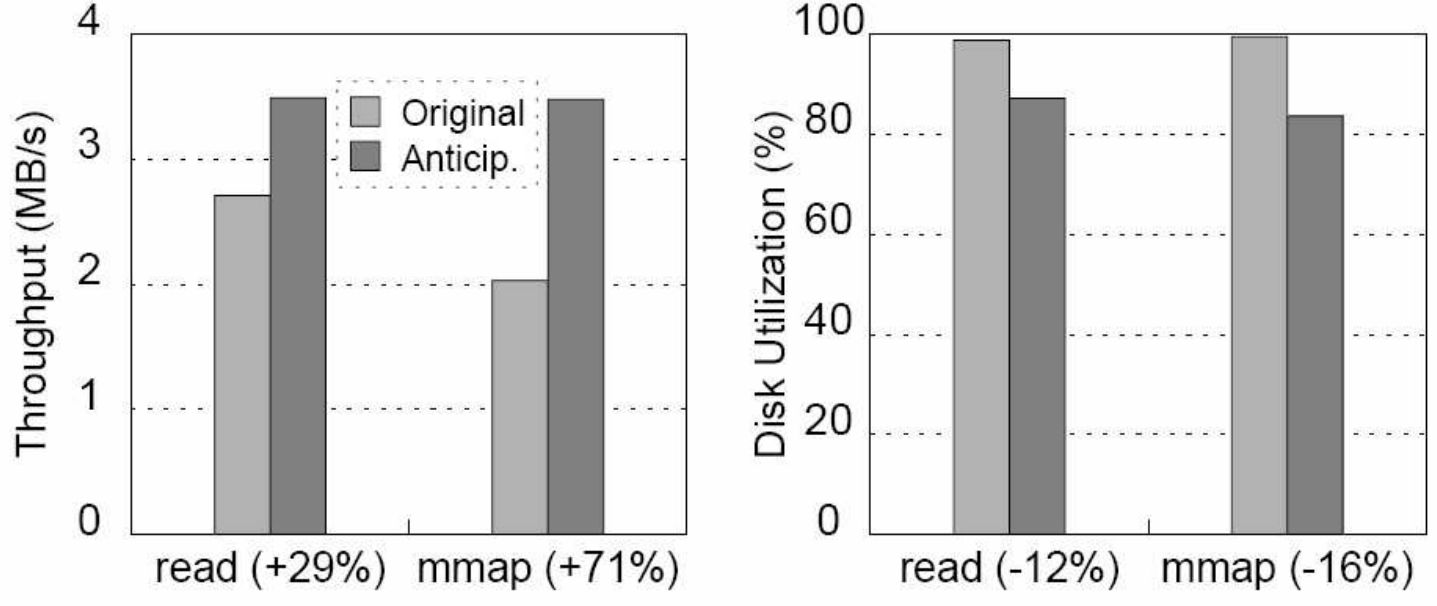


Figure 9: The Apache Webserver configured in two modes, read and mmap. The former exemplifies the practical limitations of filesystem prefetch.

Question

- It is not possible that delayed disk schedule merges synchronous read operations.