16th USENIX Conference on File and Storage Technologies (FAST'18 WiP)
OAKLAND, CA, USA, FEBRUARY 12–15, 2018

Fair Down to the Device:

A GC-aware Fair Scheduler for SSD

Lun Wang*, Cheng Ji†, Liang Shi‡, Chia-Lin Yang* and Chun Jason Xue†

*National Taiwan University
†City University of Hong Kong
‡Chongqing University

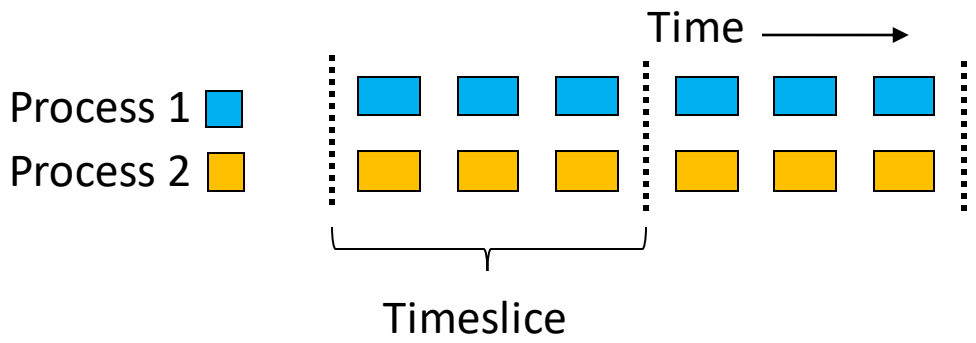






Fairness vs. Garbage Collection (GC)

- **Fair I/O scheduler on SSD** [FAST'12, ATC'13]
  - FIOS [Park et al., FAST'12]
    - Each process is assigned with equal timeslice and multiple processes are allowed to access SSD simultaneously



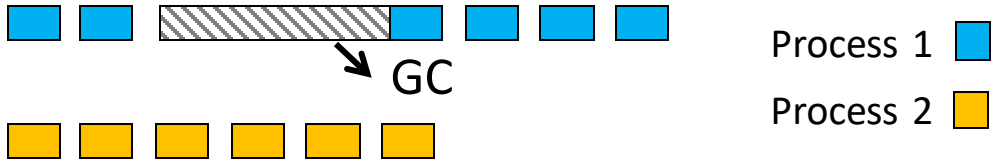
Fairness Metric:

$$I/O\ slowdown\ ratio = \frac{Avg\ I/O\ latency_{\ concurrent}}{Avg\ I/O\ latency_{\ alone}}$$

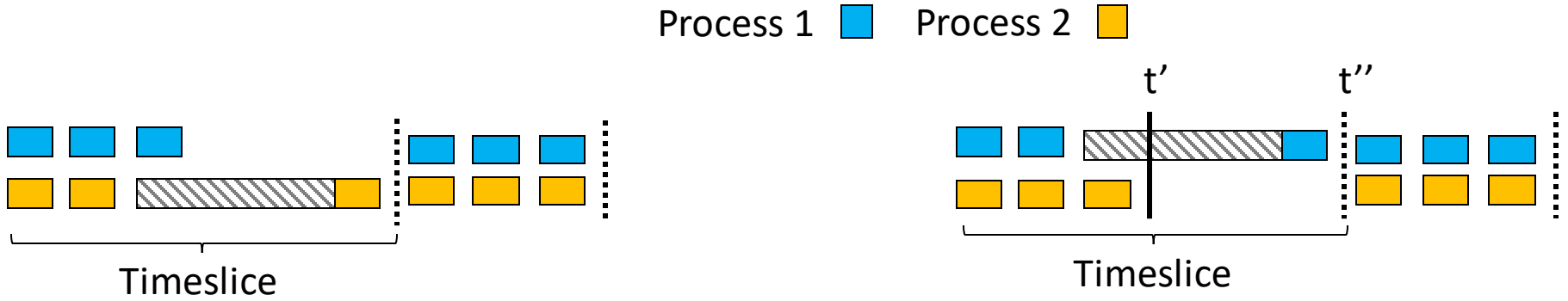
For *N* concurrent processes, *I/O slowdown ratio* should be no larger than *N*

Fairness vs. Garbage Collection (GC)

- Running alone



- Running concurrently



Scenario 1

GC penalty is unfairly attributed to process 2

Scenario 2

Process 2 is indirectly blocked by GC since its timeslice is refreshed at t'' even though it consumes the entire timeslice at t'

Our Contribution

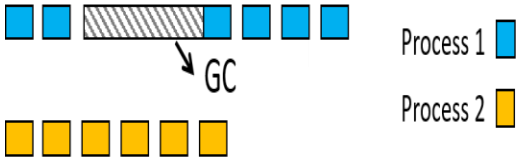
The first I/O scheduler to achieve fairness under GC for SSDs



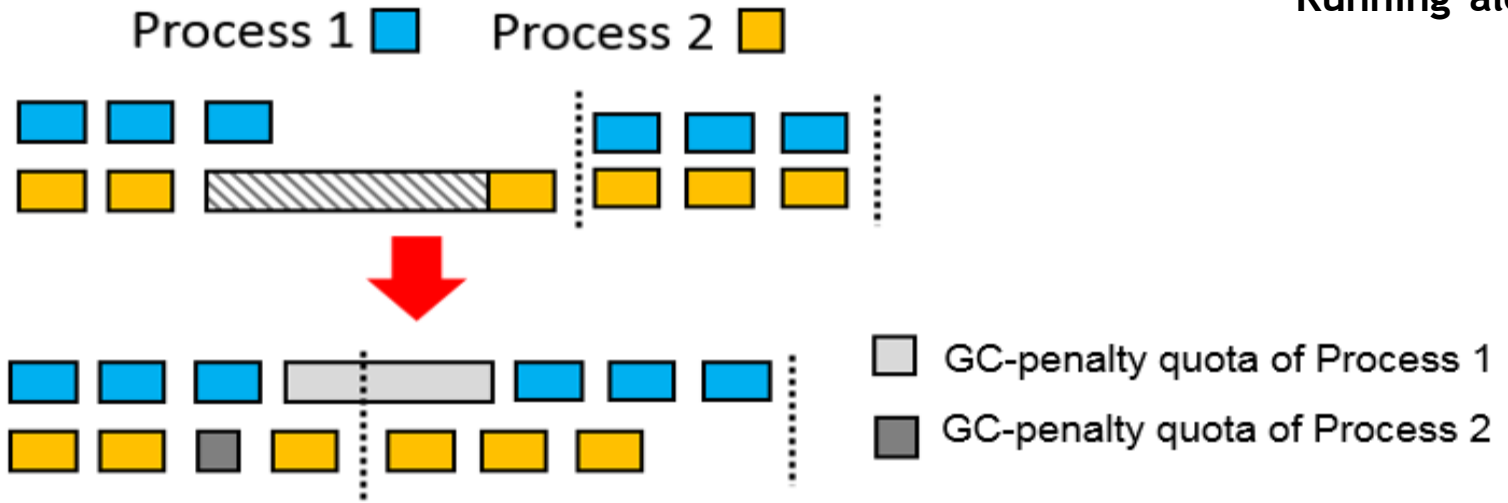
To ensure fairness under GC, a process's performance should be penalized by GC in the same degree as when it runs alone

GC-Aware Scheduler: Design Principle

- Orchestrate GC operations based on a new concept called **GC-penalty quota**
  - Time that a process could be blocked by GC in an epoch



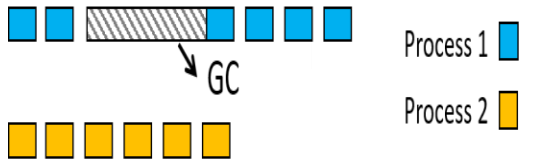
Running alone



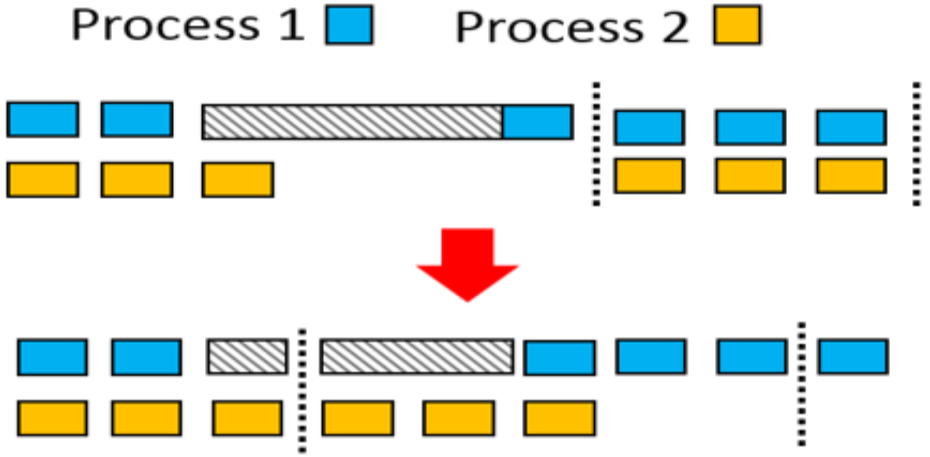
Scenario 1

GC-Aware Scheduler: Design Principle

- Timeslice should be updated in a timely manner via a new signal from the device to the host when encountering GC

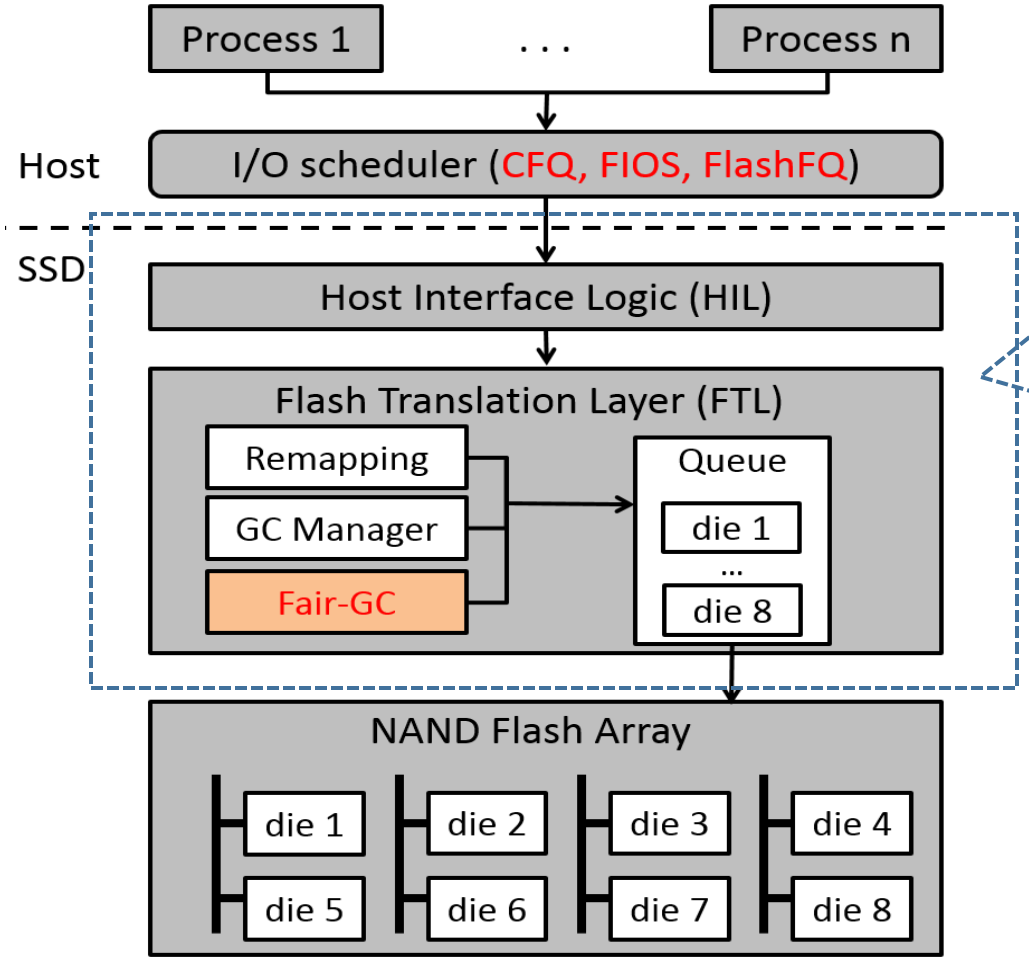


Running alone

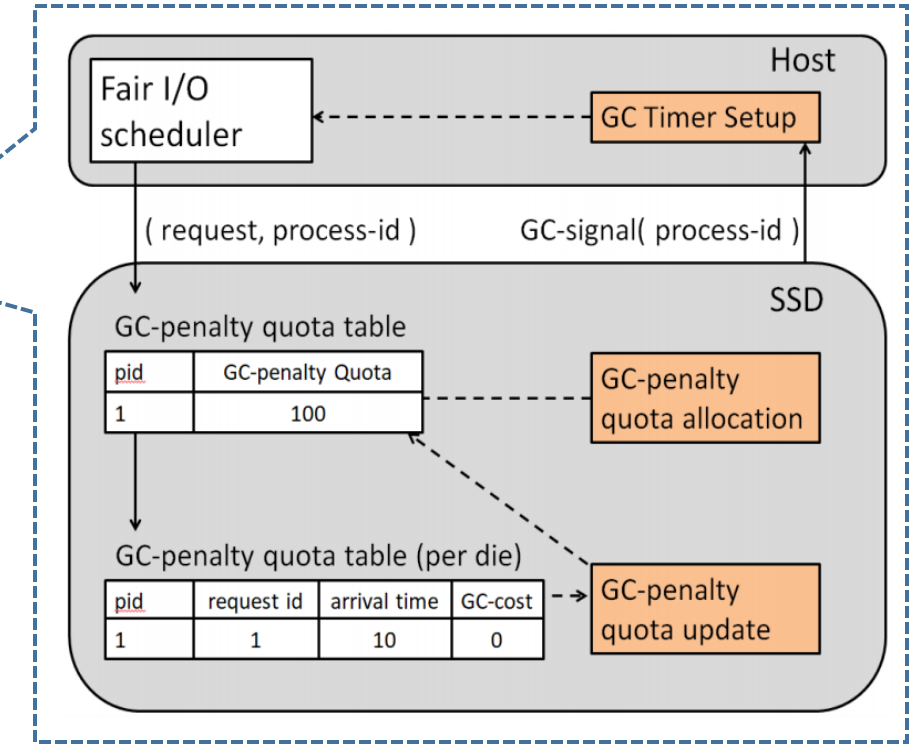


Scenario 2

GC-Aware Scheduler: Mechanism

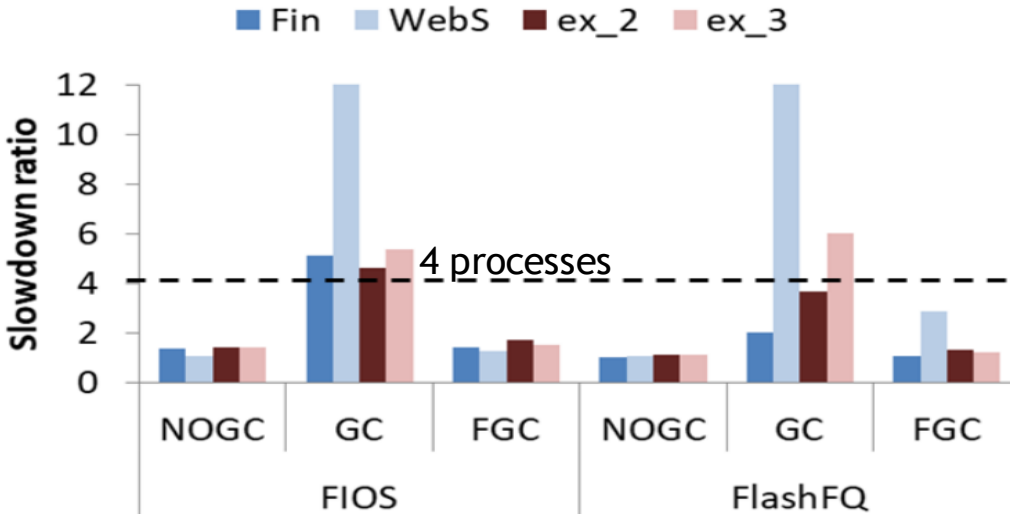


Overall SSD architecture with Fair-GC

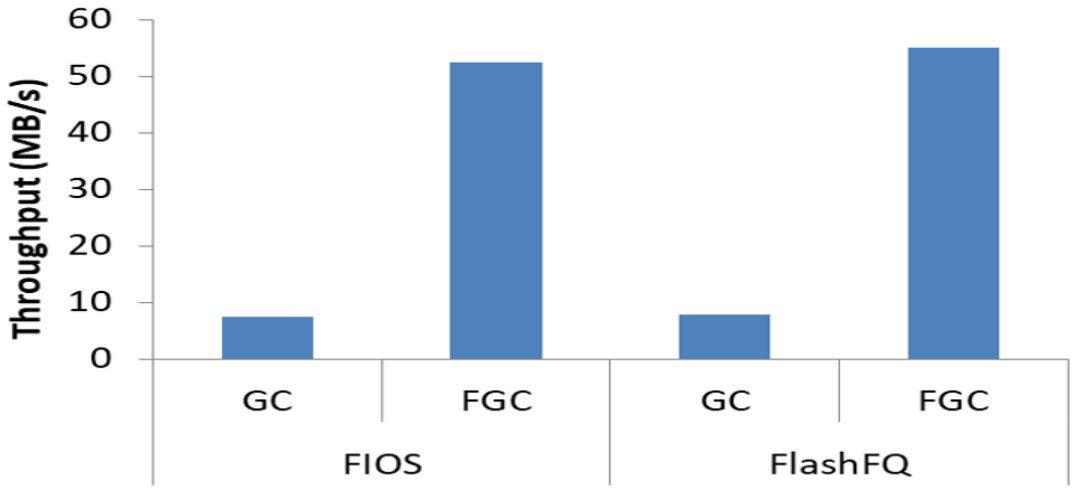


Design overview of Fair-GC scheduler

Evaluation



Both FIOS and FlashFQ are able to achieve fairness with Fair-GC (FGC)



FGC also improves throughput by eliminating the SDD inefficiency due to the GC

Questions?