When Poll is More Energy Efficient than Interrupt

Bryan Harris and Nihat Altiparmak

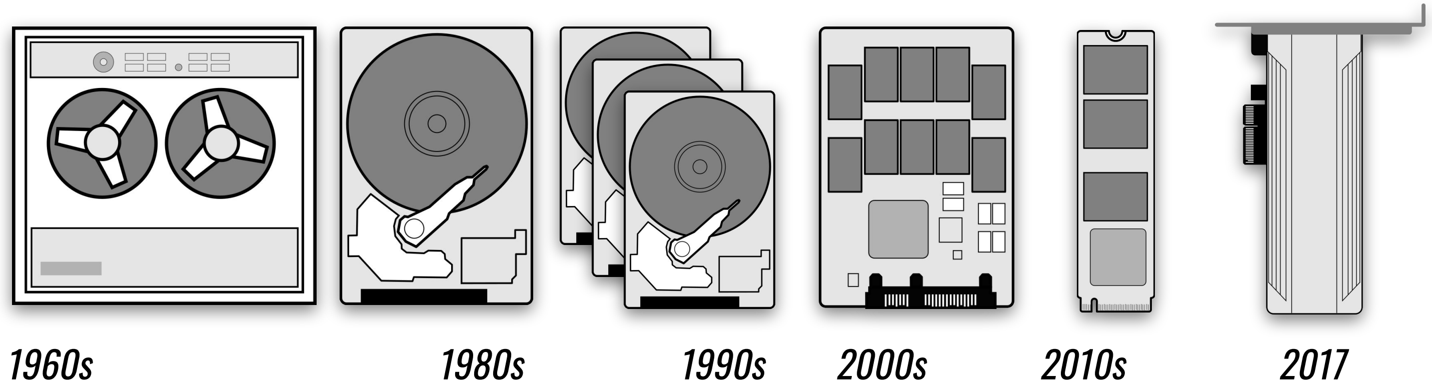
Computer Systems Lab

Computer Science & Engineering Department

UNIVERSITY OF LOUISVILLE®

June 27, 2022

Trends in data storage



***Magnetic** media* | ***Flash** media* | *New technologies*

Fully sequential access → Highly sequential → Parallel devices → High internal parallelism

Moving parts ← | → Fully electronic

Changing mechanics

Performance prioritized

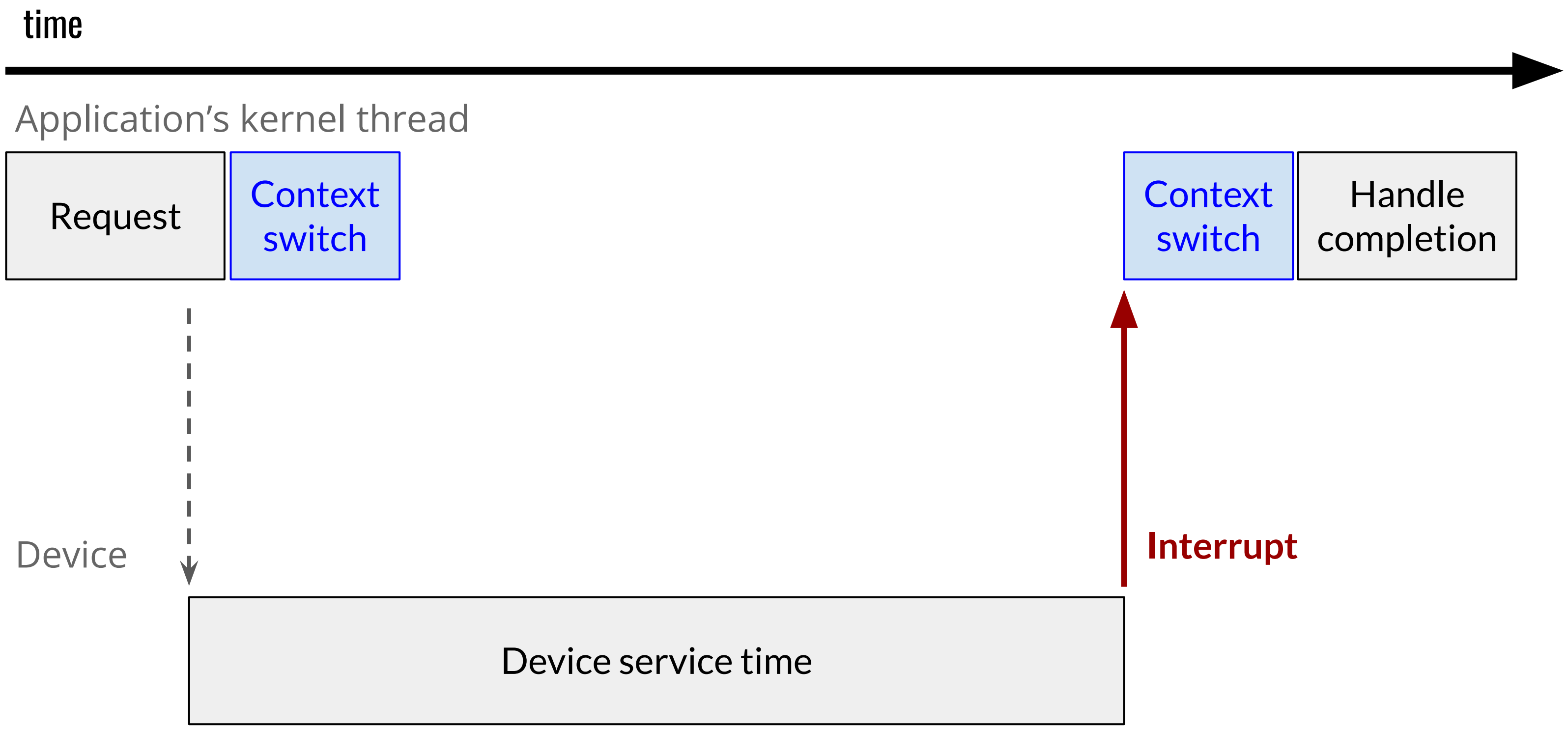
Capacity prioritized

Energy considered

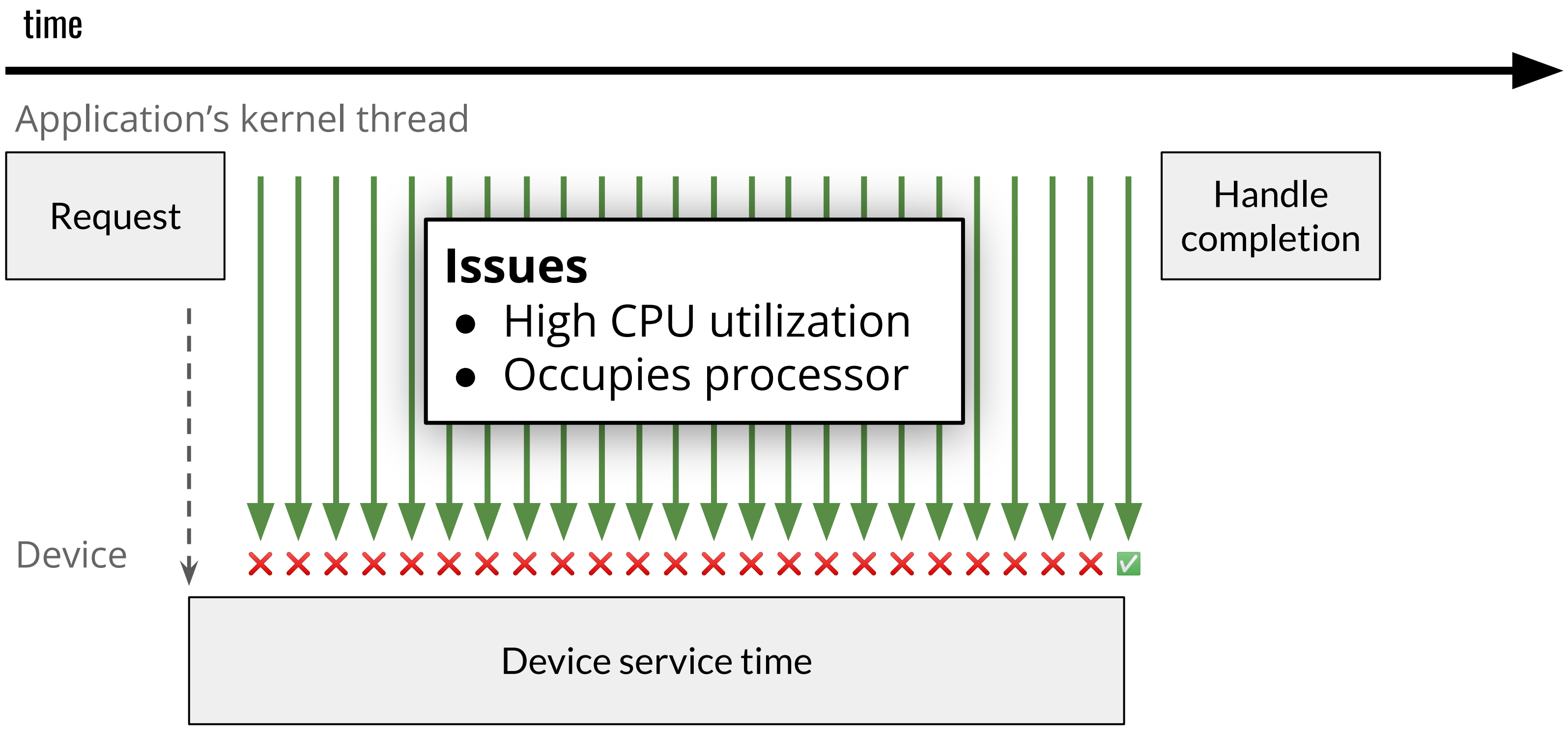
Changing priorities

IO Completion

Interrupt


time
Application's kernel thread
Request
Context switch
Context switch
Handle completion
Device
Interrupt
Device service time

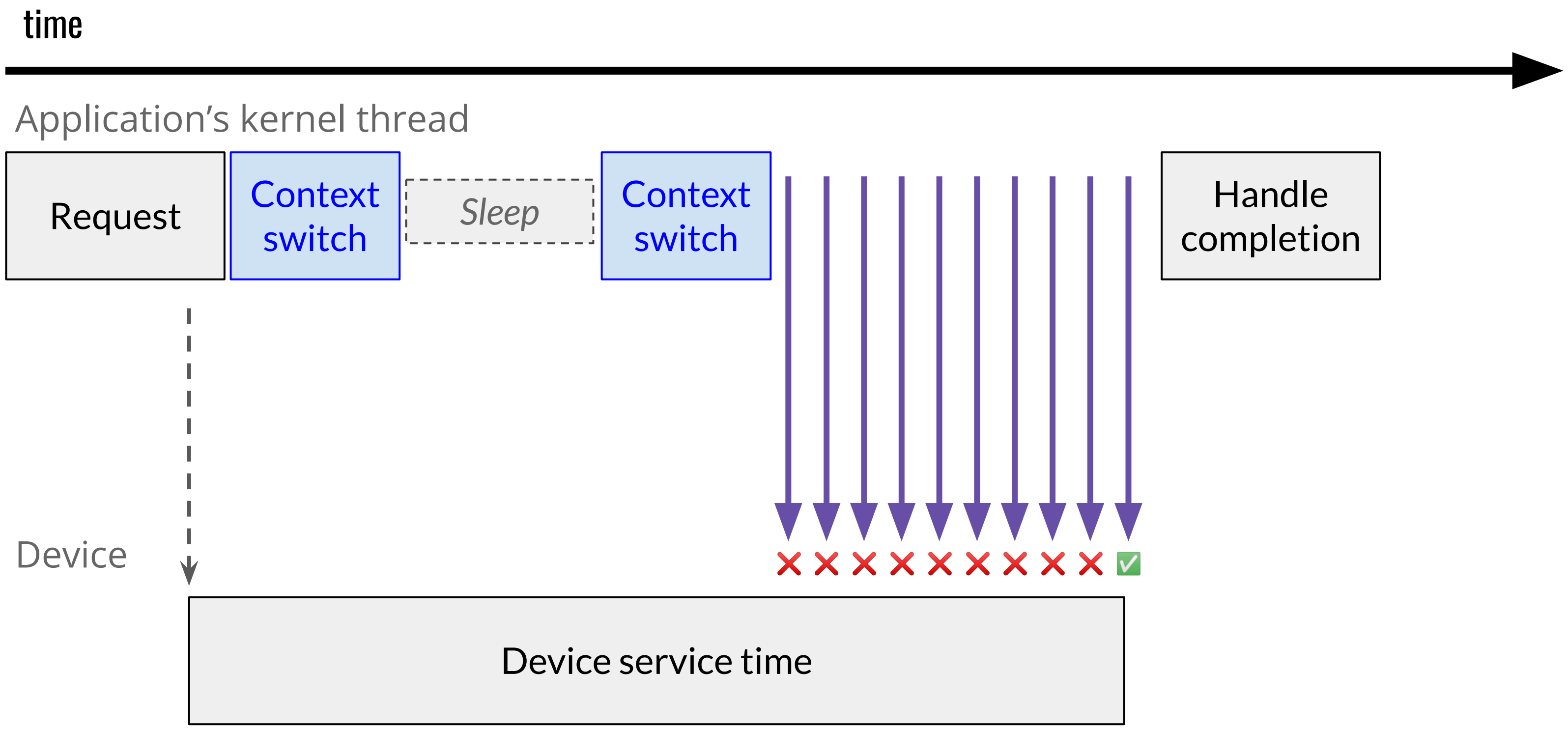
Polling (Classic)



Issues
- High CPU utilization
- Occupies processor

UNIVERSITY OF LOUISVILLE

Polling (Hybrid)

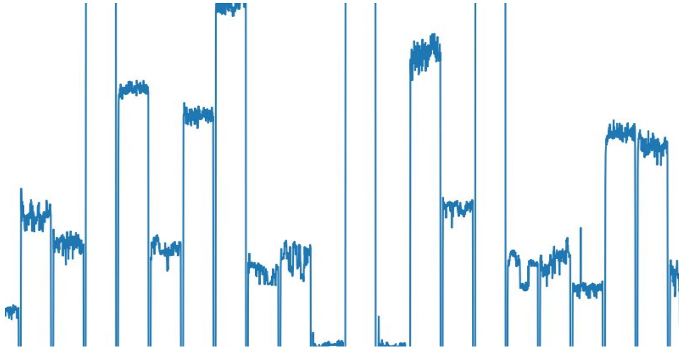


Related Work

- **“When Poll is Better than Interrupt”**
  by Yang, Minturn, and Hady *(FAST ’12)*
- **“Reducing DRAM Footprint with NVM in Facebook”**
  by Eisenman et al. *(EuroSys ’18)*
- **“FlashShare: Punching Through Server Storage Stack from Kernel to Firmware for Ultra-Low Latency SSDs”**
  by Zhang et al. *(OSDI ’18)*
- Polling also supported in *io_uring* and *SPDK*

Experimental setup

Image from Onset Computer Corporation



Power Measurement
Onset HOBO plug meter logs power, current, etc., every second, for the entire system.

Workloads
fio ("Flexible IO tester")

- *preadv2*
  - O_DIRECT
  - RWF_HIPRI (polling)
- *ext4* file system
- "none" IO scheduler

Range of request sizes and threads

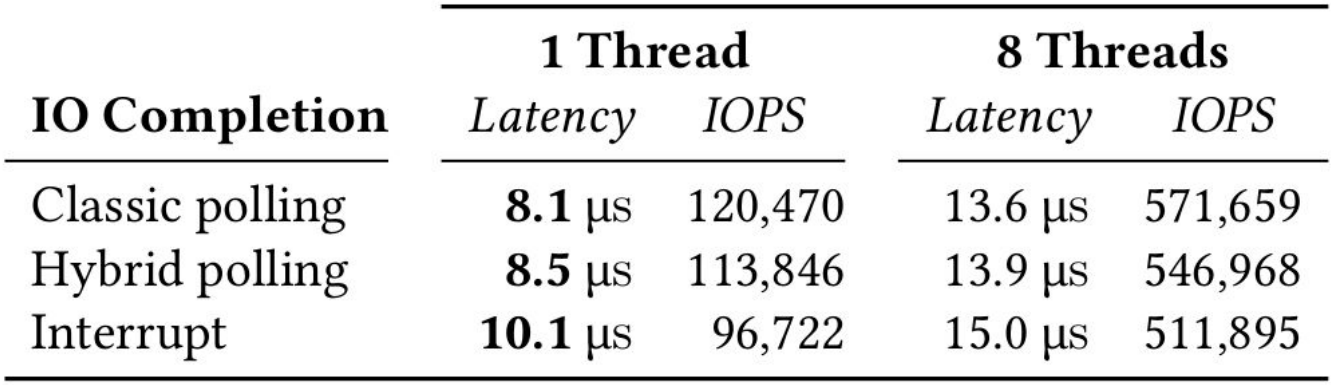
Goals of Polling

Observation 1

Both classic and hybrid polling show **improved performance** over interrupts.

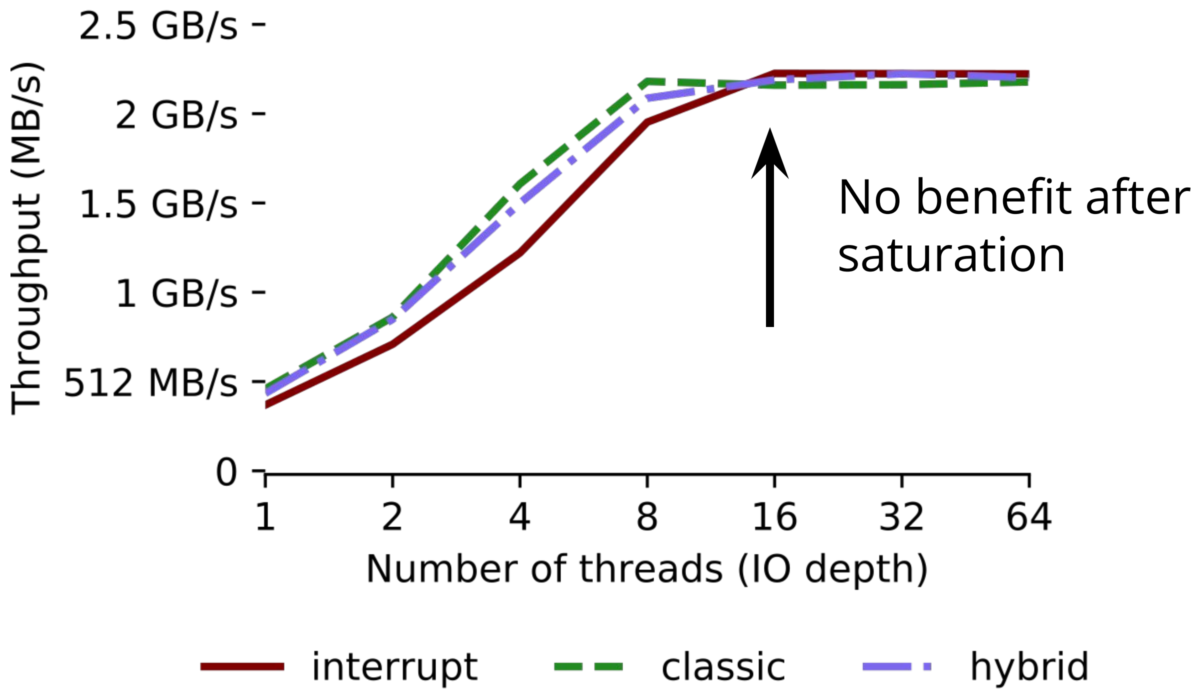
- Shorter latency
- Higher throughput

Obs. 1: Polling improves performance



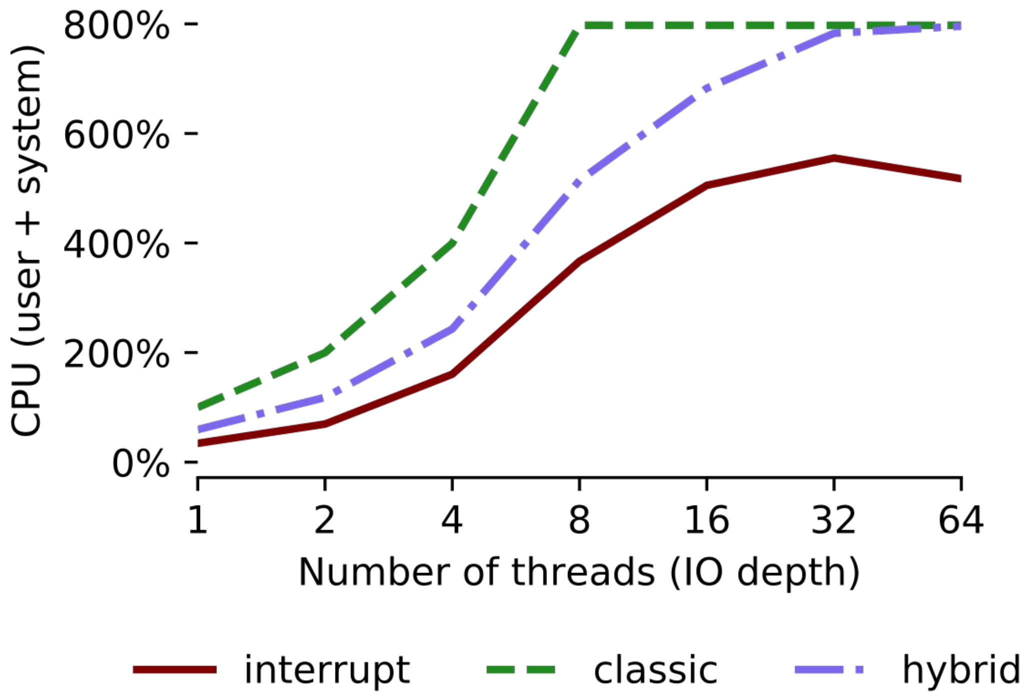
IO Completion	**1 Thread**		**8 Threads**	
	Latency	*IOPS*	*Latency*	*IOPS*
Classic polling	**8.1** µs	120,470	13.6 µs	571,659
Hybrid polling	**8.5** µs	113,846	13.9 µs	546,968
Interrupt	**10.1** µs	96,722	15.0 µs	511,895

Obs. 1: Polling improves performance



UNIVERSITY OF LOUISVILLE.

Obs. 2: Hybrid polling reduces CPU usage over classic polling

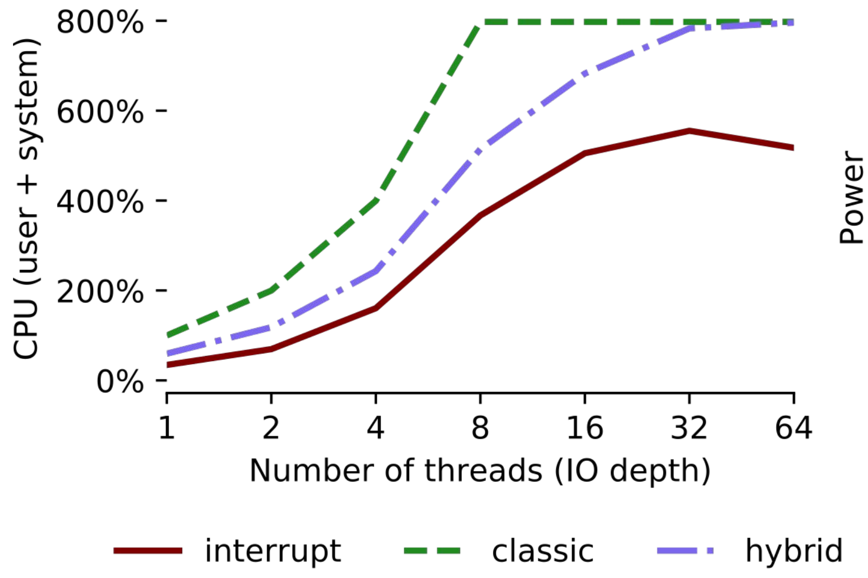


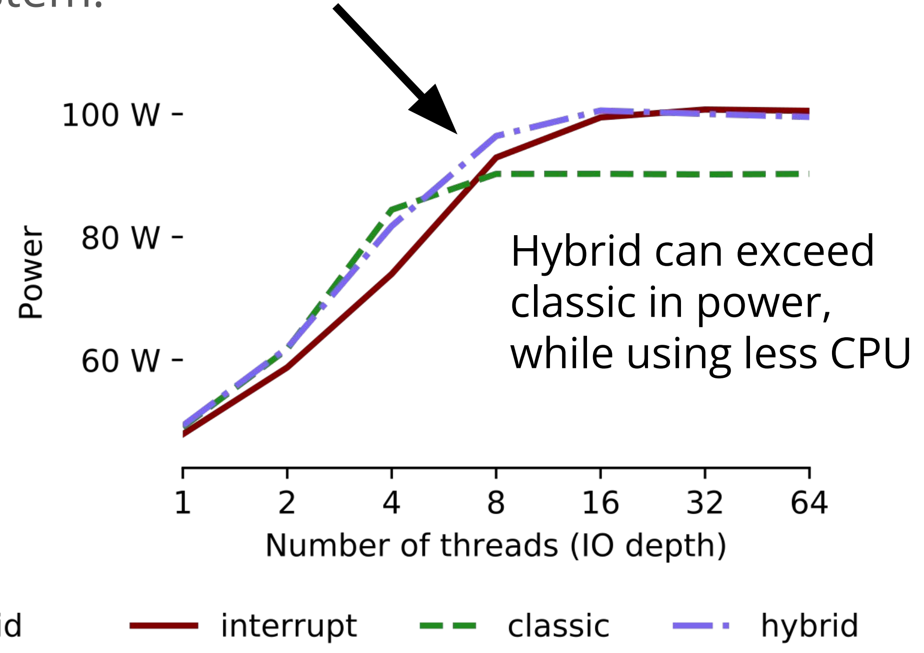
UNIVERSITY OF LOUISVILLE.

Costs of Polling

Observation 3

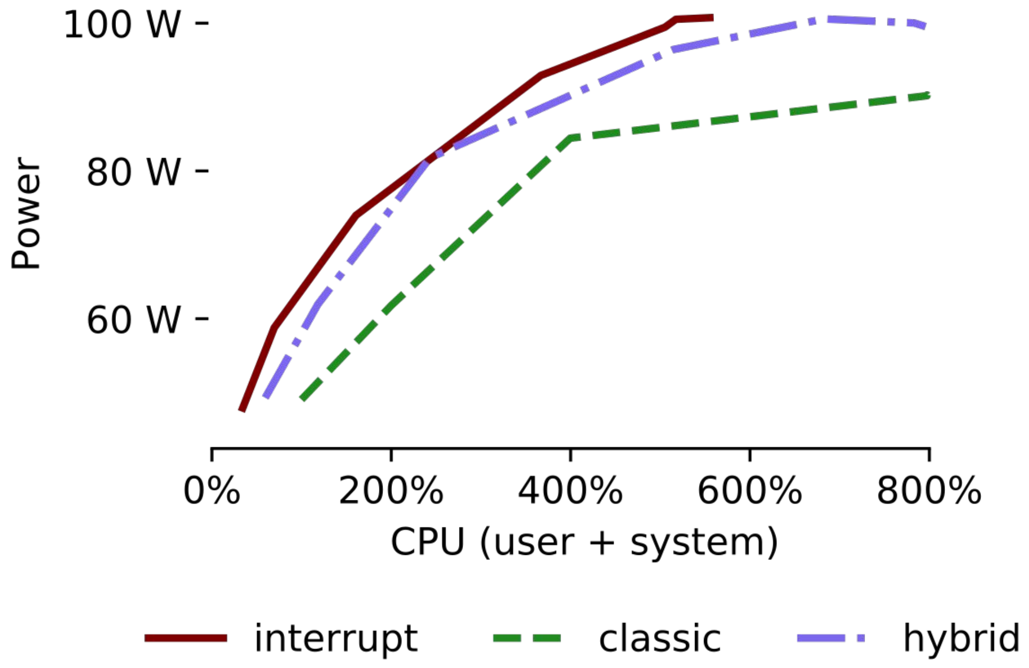
CPU utilization does not directly correspond to power consumption of the entire system.





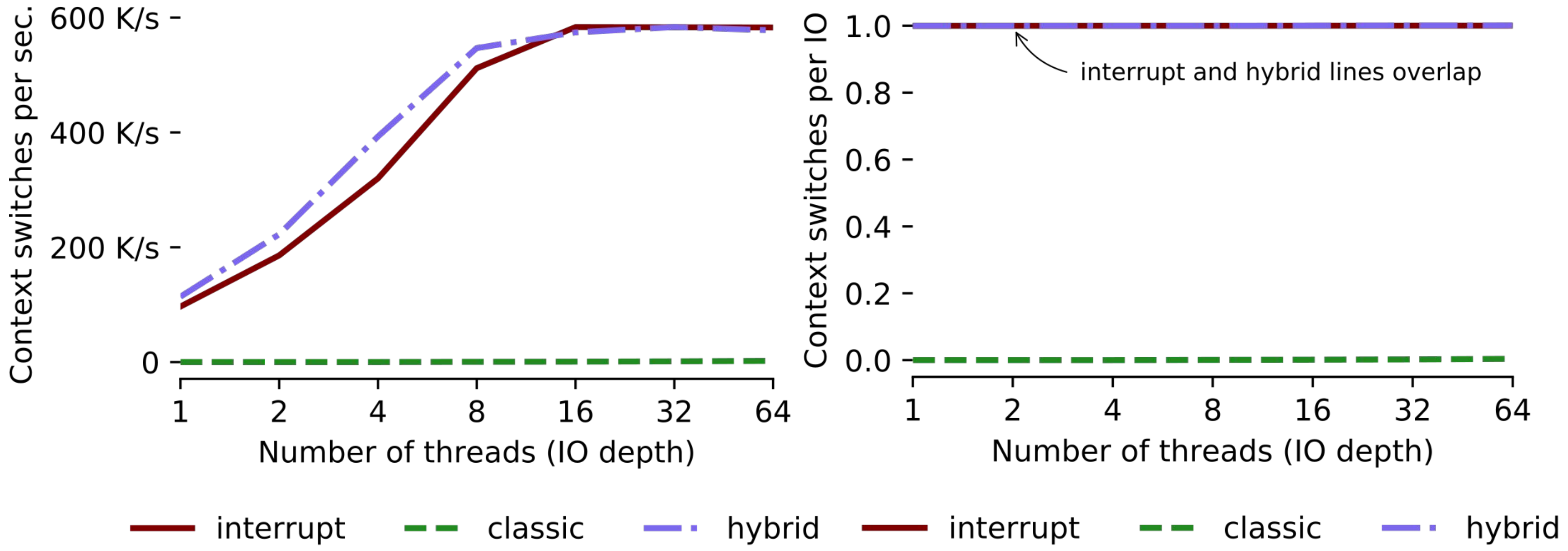
Hybrid can exceed classic in power, while using less CPU

Obs. 3: CPU does not directly correspond with power consumption



Observation 4

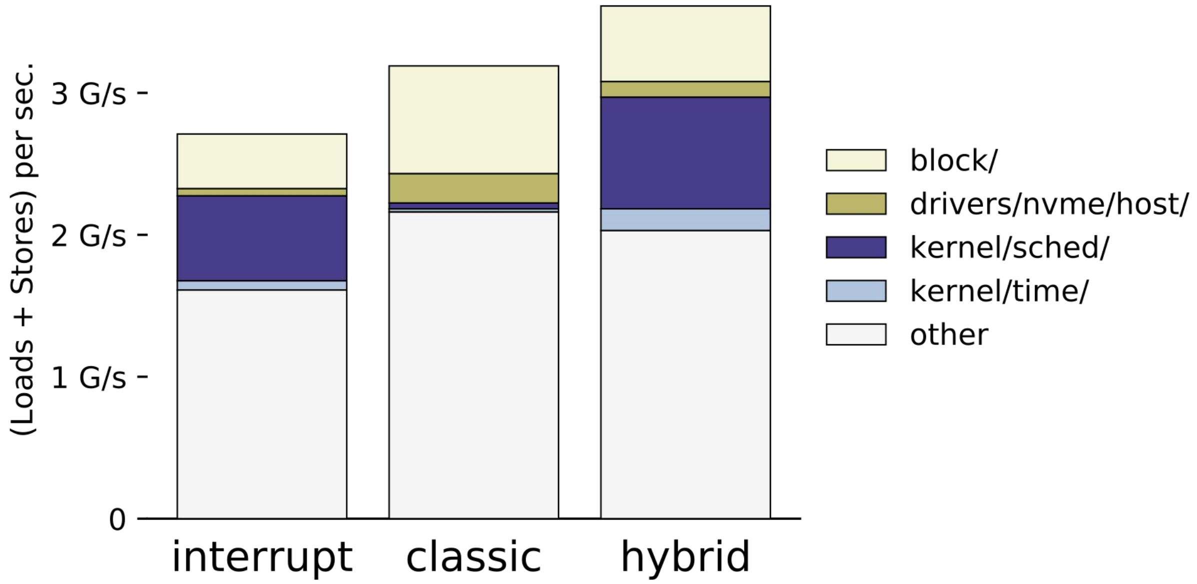
Hybrid polling triggers as many **context switches** as interrupts, while classic polling triggers none.



UNIVERSITY OF LOUISVILLE.

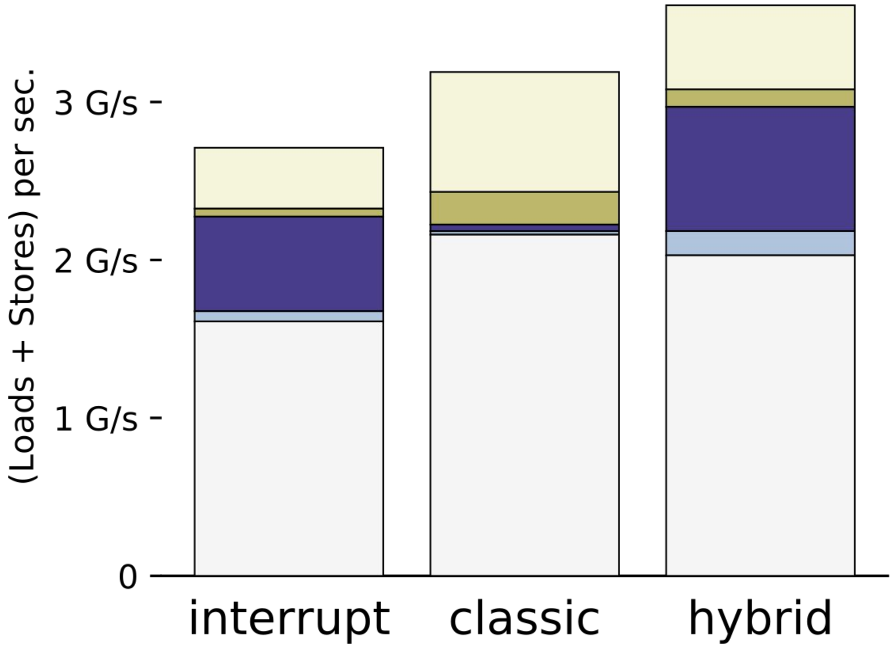
Observation 5

Hybrid polling has a high cost in **load/store operations** associated with context switches.



UNIVERSITY OF LOUISVILLE

Obs. 5: Load/store costs

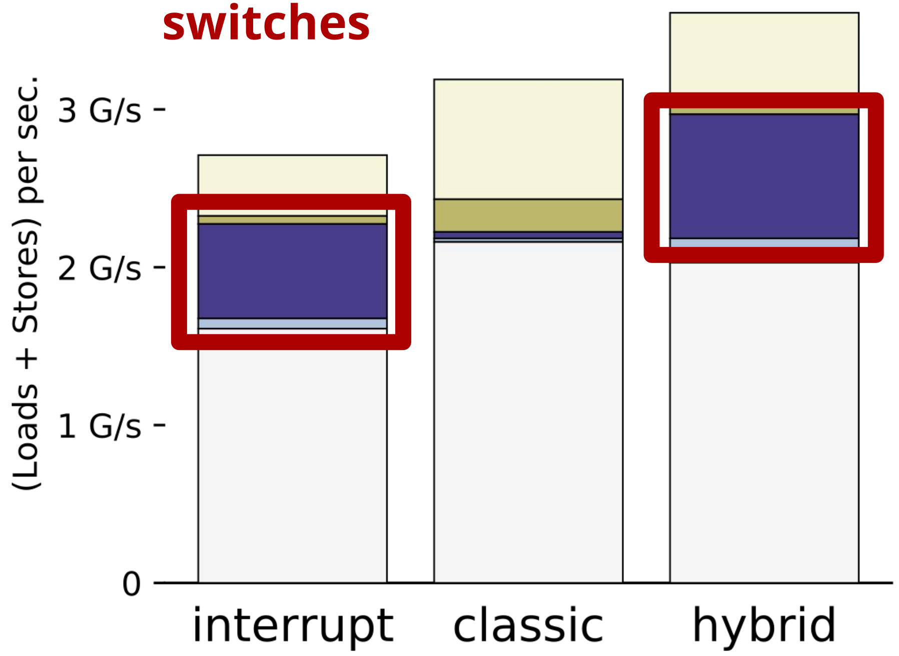


Source	Interrupt	Classic	Hybrid
⬜ block/	386	759	533
blk_poll	*0*	*219*	*70*
blk_mq_* *(36 functions)*	*111*	*158*	*156*
(109 other functions)	*274*	*383*	*307*
🟨 drivers/nvme/host/	51	207	111
🟪 kernel/sched/	598	41	786
__schedule	*13*	*0*	*13*
psi_task_change	*11*	*0*	*13*
update_load_avg	*14*	*0*	*22*
(285 other functions)	*560*	*41*	*739*
🟦 kernel/time/	65	22	154
ktime_get	*10*	*14*	*22*
hrtimer *(33 functions)*	*0*	*1*	*100*
(56 other functions)	*55*	*8*	*33*
⬜ *(other sources)*	1611	2160	2029
Total	**2710** M/s	**3190** M/s	**3612** M/s

UNIVERSITY OF LOUISVILLE

Obs. 5: Load/store costs

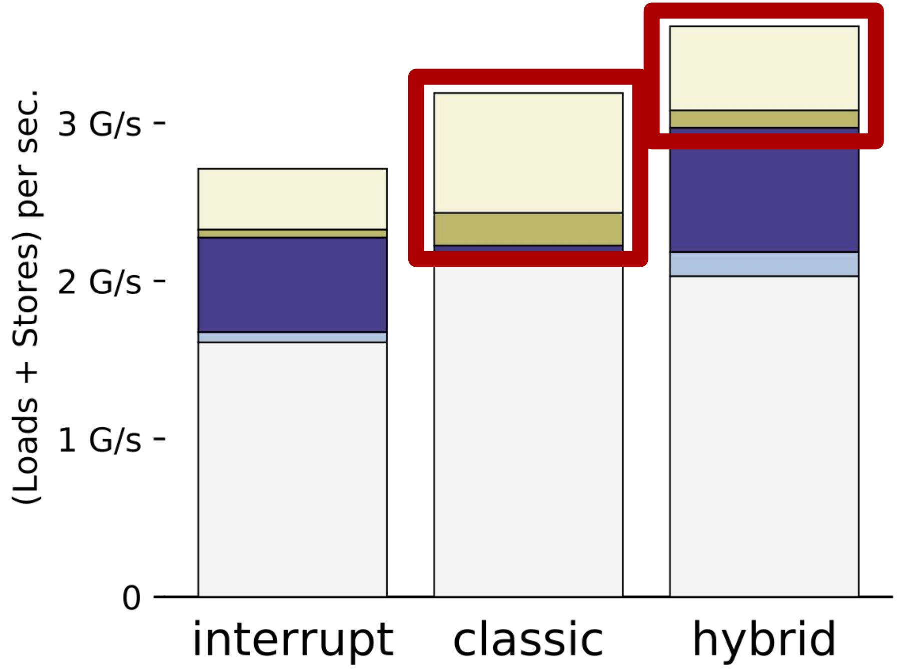
CPU scheduling due to context switches

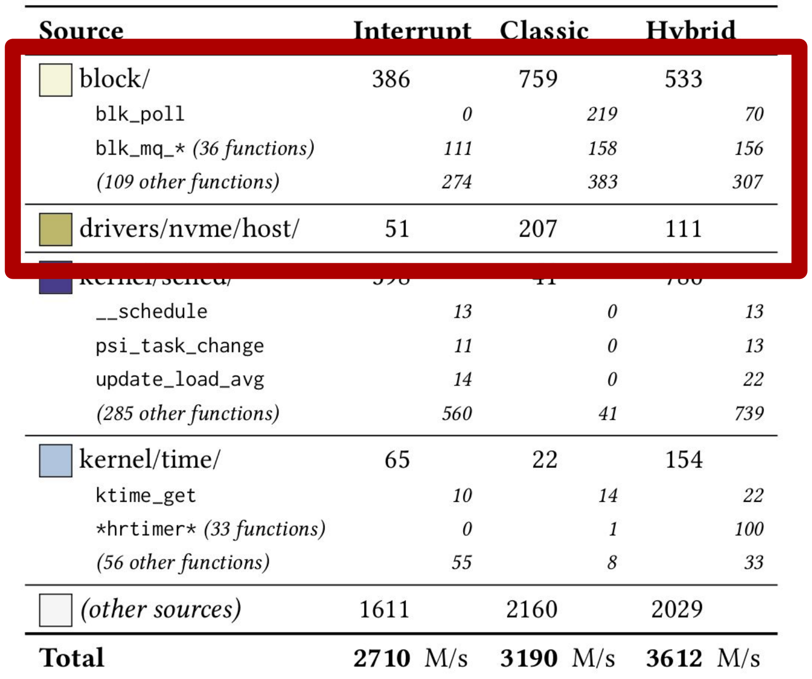


Source	Interrupt	Classic	Hybrid
block/	386	759	533
blk_poll	*0*	*219*	*70*
blk_mq_* *(36 functions)*	*111*	*158*	*156*
(109 other functions)	*274*	*383*	*307*
drivers/nvme/host/	51	207	111
kernel/sched/	598	41	786
__schedule	*13*	*0*	*13*
psi_task_change	*11*	*0*	*13*
update_load_avg	*14*	*0*	*22*
(285 other functions)	*560*	*41*	*739*
kernel/time/	65	22	154
ktime_get	*10*	*14*	*22*
hrtimer *(33 functions)*	*0*	*1*	*100*
(56 other functions)	*55*	*8*	*33*
(other sources)	1611	2160	2029
Total	**2710** M/s	**3190** M/s	**3612** M/s

Obs. 5: Load/store costs

Loads due to polling

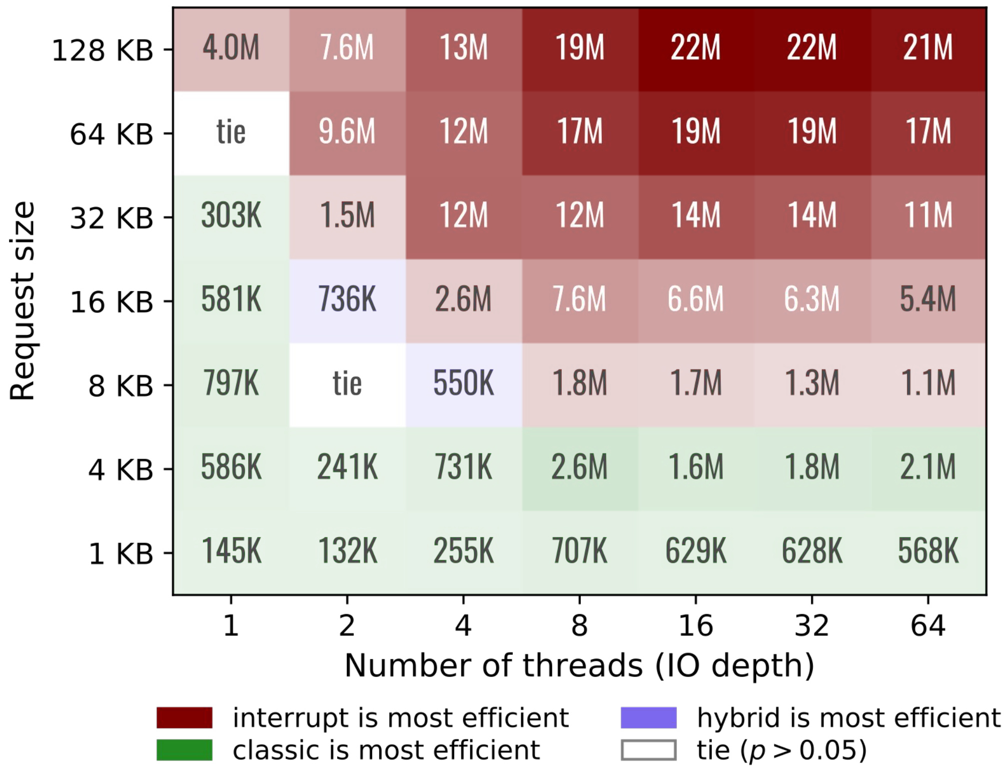




Source	Interrupt	Classic	Hybrid
▨ block/	386	759	533
blk_poll	*0*	*219*	*70*
blk_mq_* *(36 functions)*	*111*	*158*	*156*
(109 other functions)	*274*	*383*	*307*
▨ drivers/nvme/host/	51	207	111
▨ kernel/sched/	598	41	786
__schedule	*13*	*0*	*13*
psi_task_change	*11*	*0*	*13*
update_load_avg	*14*	*0*	*22*
(285 other functions)	*560*	*41*	*739*
▨ kernel/time/	65	22	154
ktime_get	*10*	*14*	*22*
hrtimer *(33 functions)*	*0*	*1*	*100*
(56 other functions)	*55*	*8*	*33*
▨ *(other sources)*	1611	2160	2029
Total	**2710** M/s	**3190** M/s	**3612** M/s

UNIVERSITY OF LOUISVILLE

Observation 6



Request size	1	2	4	8	16	32	64
128 KB	4.0M	7.6M	13M	19M	22M	22M	21M
64 KB	tie	9.6M	12M	17M	19M	19M	17M
32 KB	303K	1.5M	12M	12M	14M	14M	11M
16 KB	581K	736K	2.6M	7.6M	6.6M	6.3M	5.4M
8 KB	797K	tie	550K	1.8M	1.7M	1.3M	1.1M
4 KB	586K	241K	731K	2.6M	1.6M	1.8M	2.1M
1 KB	145K	132K	255K	707K	629K	628K	568K

Polling can be **more energy efficient** than interrupts.

Conclusions

- Both polling methods achieve their design goals of improved performance
- Hybrid polling does reduce CPU usage, but not power

- Classic polling is more energy efficient than both interrupts and hybrid polling for low latency requests
- Hybrid polling has costs of both:
  - *context switching* cost associated with interrupts
  - *polling* cost associated with polling

Discussion

- ULL is so fast, consider using polling, even while valuing energy efficiency
- We expect even lower latency devices in the future

Questions?