

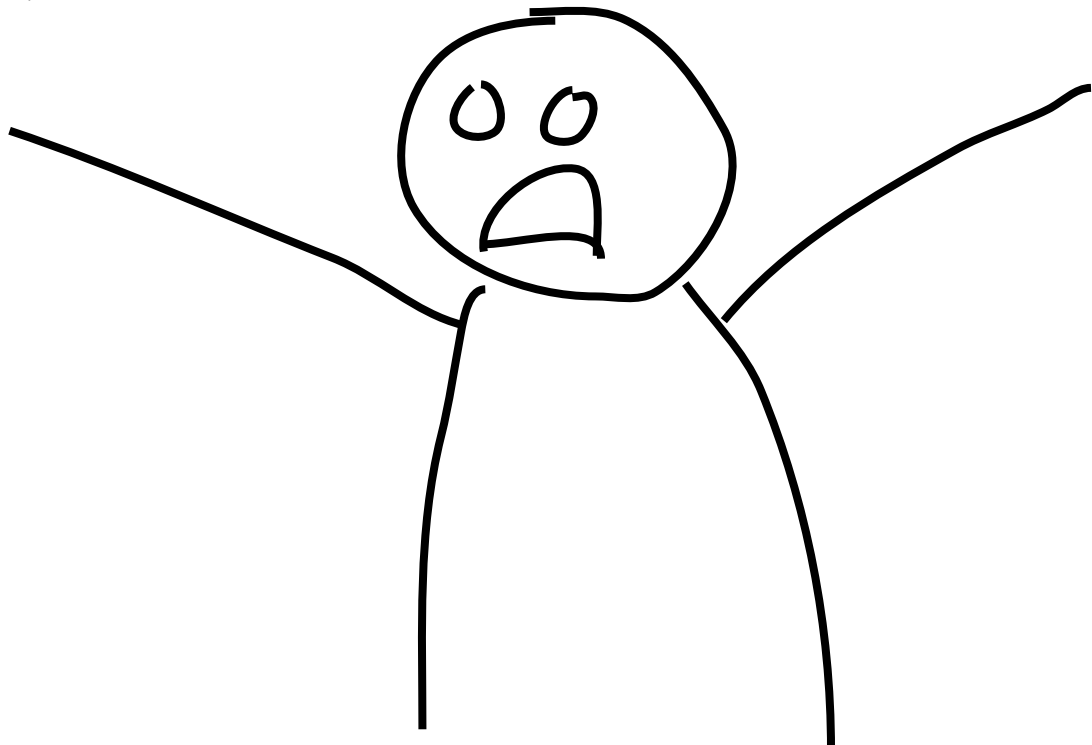
---

# The Impact of Non-Volatile Memory on Modern Computer Systems

Giulia Frascaria

---

**Now THAT's a mouthful!**



---

Let me start with some  
context...

---

# Why are we even here?

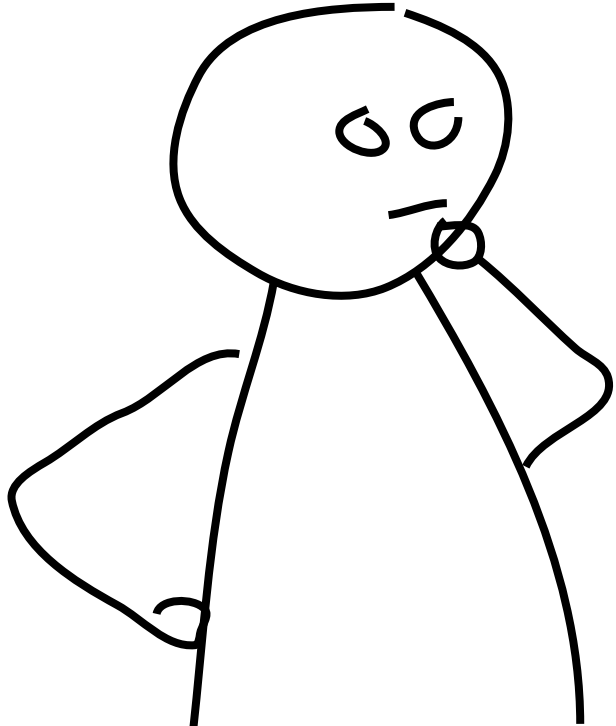
- Lack of courses on storage technologies
- Storage is a fundamental part of Computer Systems
- Evolving hardware and software should not go unnoticed

---

# What do we know?

- Traditionally, storage has been the outcast
- Avoid interaction at all costs
- Heavy performance penalty
- OS mechanisms to avoid it

***So storage is  
just a necessary evil, right?***

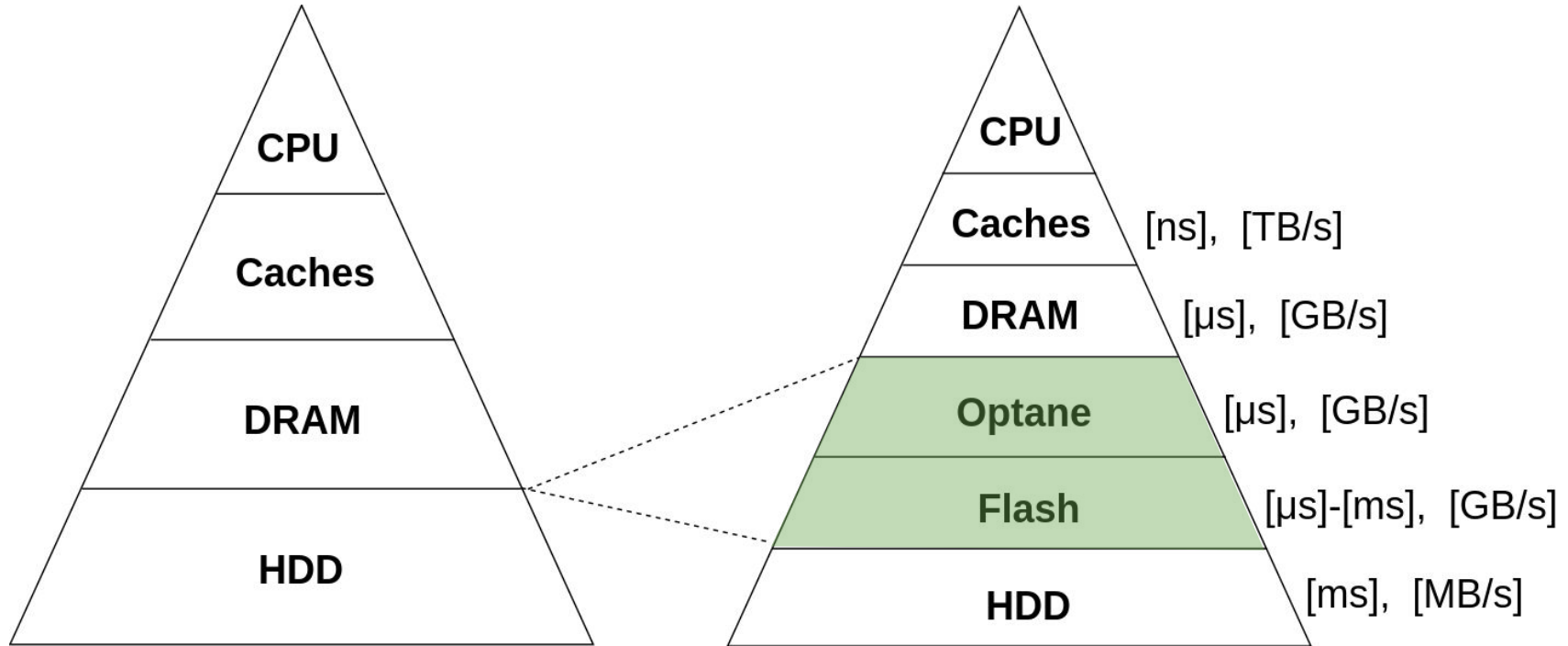


---

# Storage technologies 101

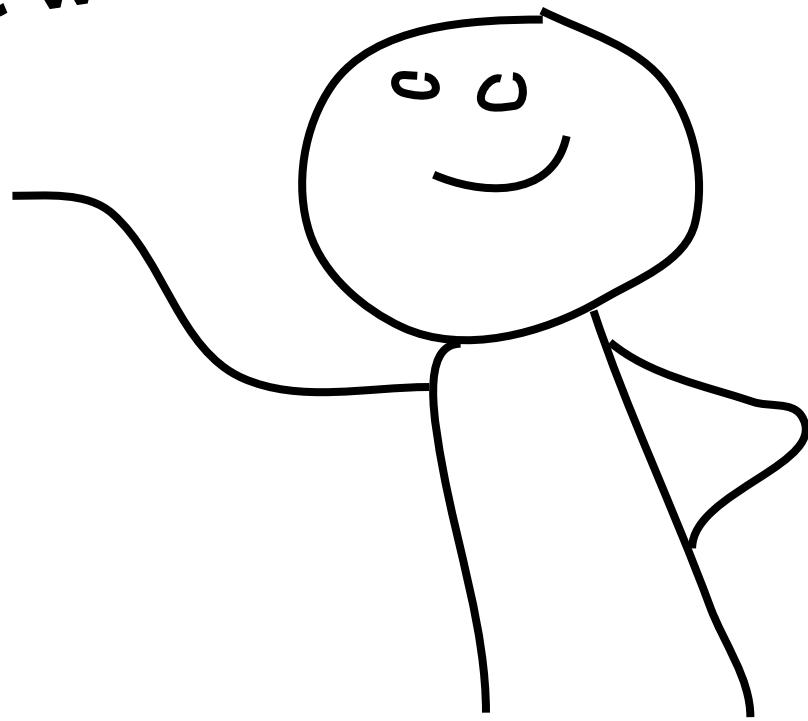
---

# The storage pyramid





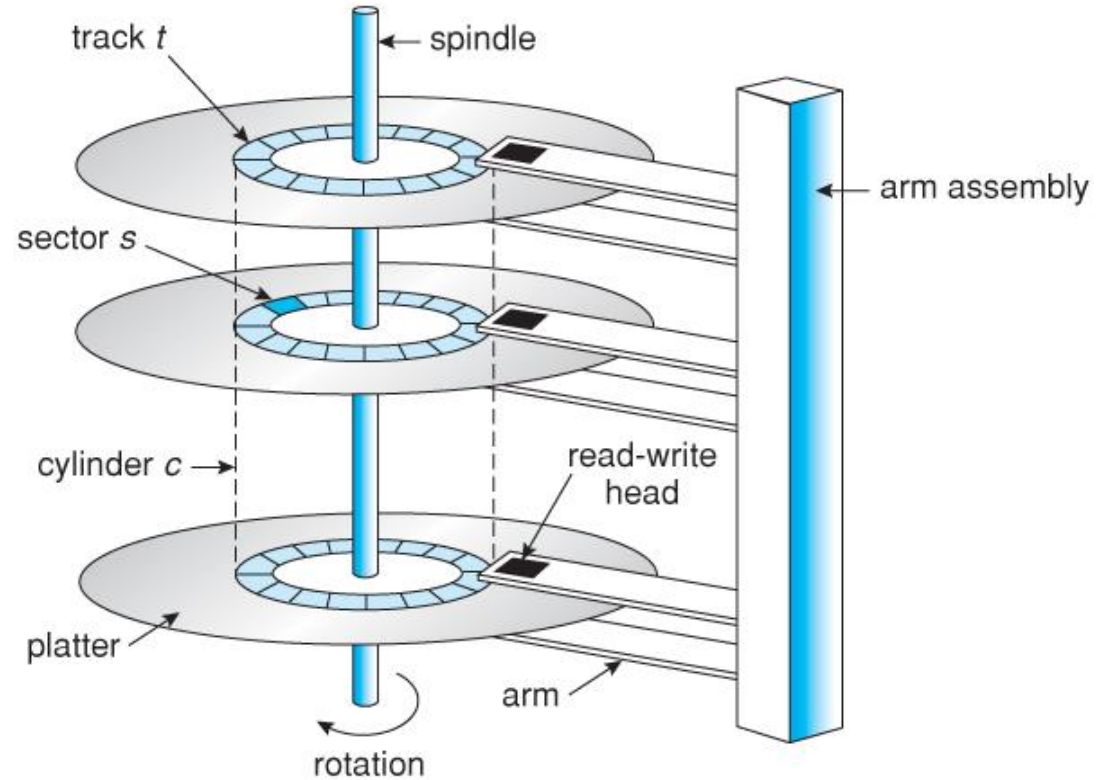
**Amazing, so the software  
will just work faster now!**



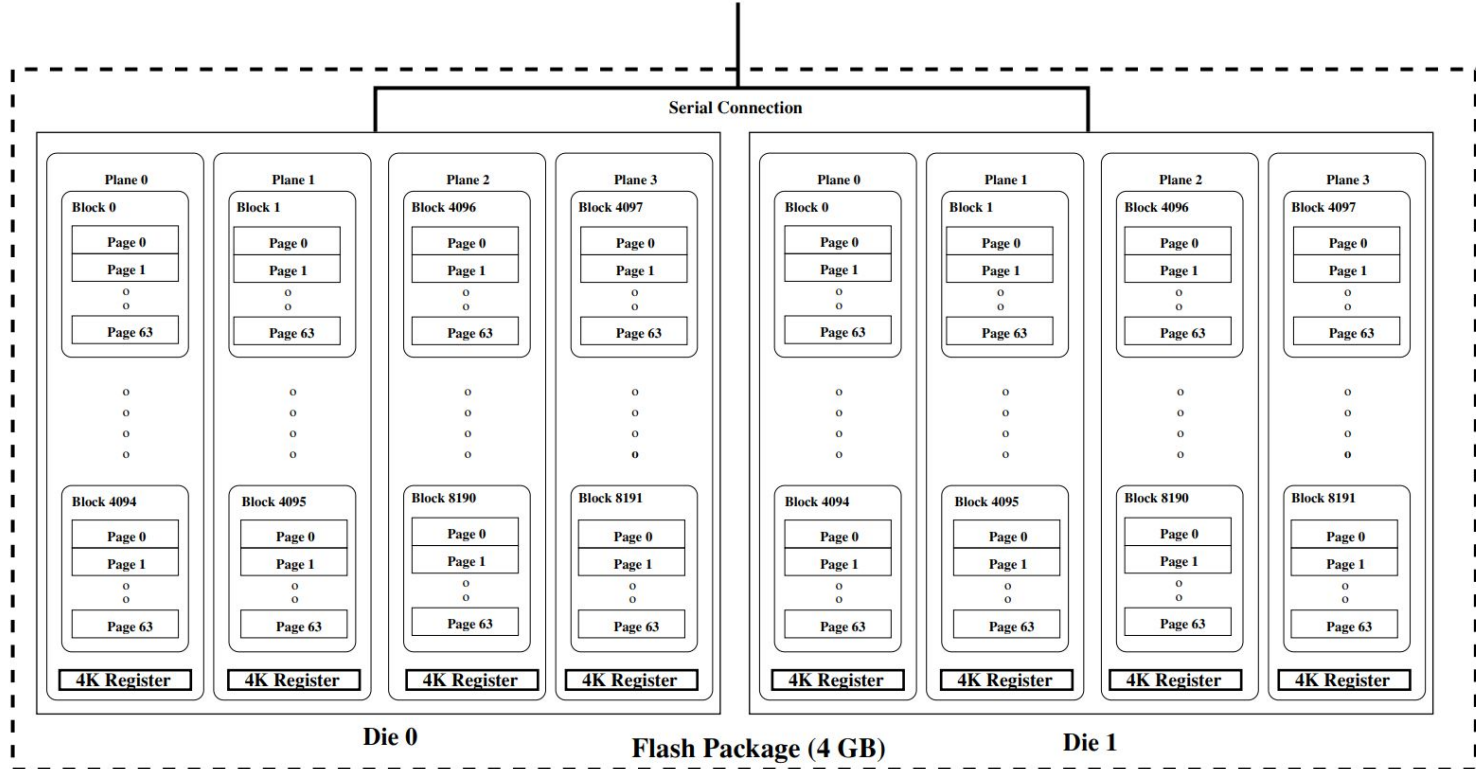
---

**Not at all!**

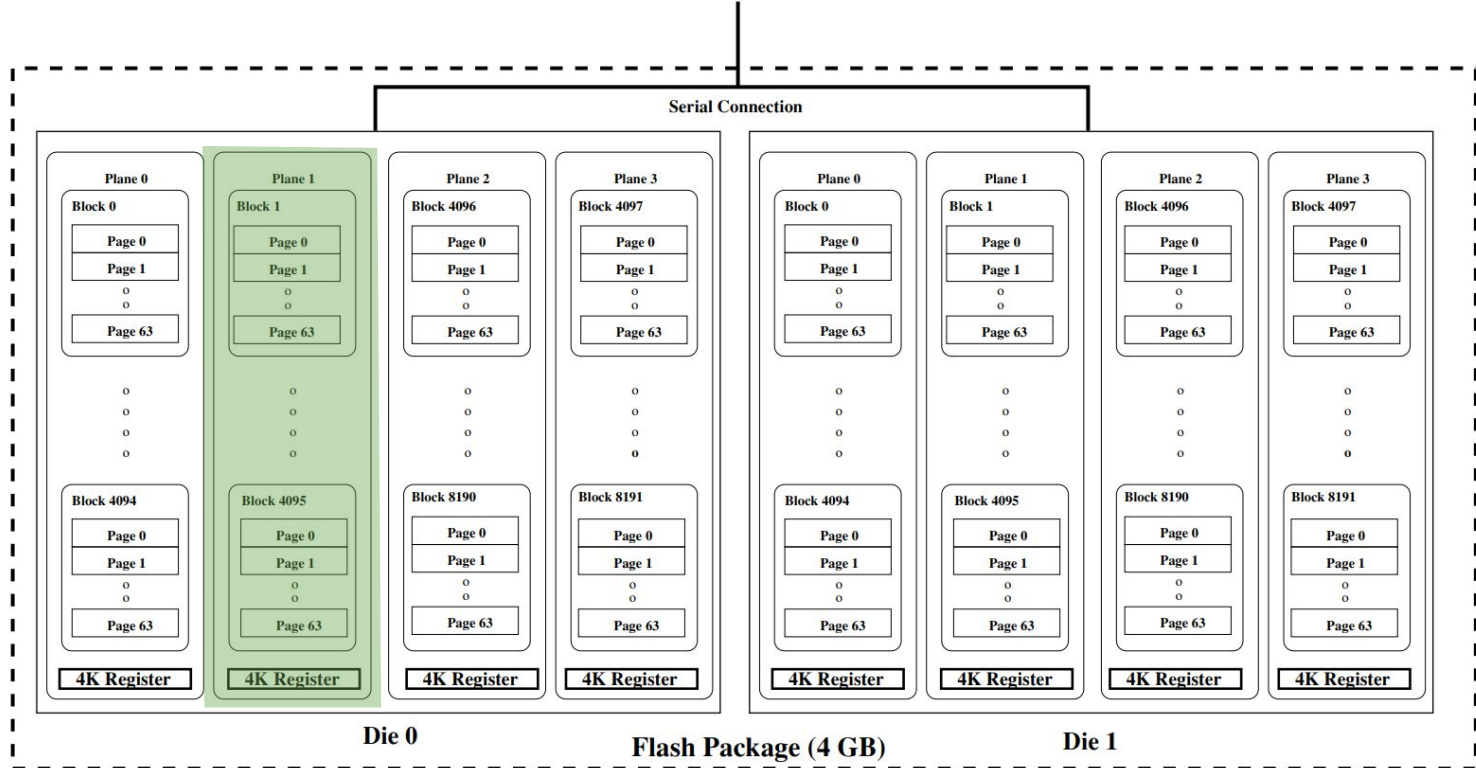
# Before



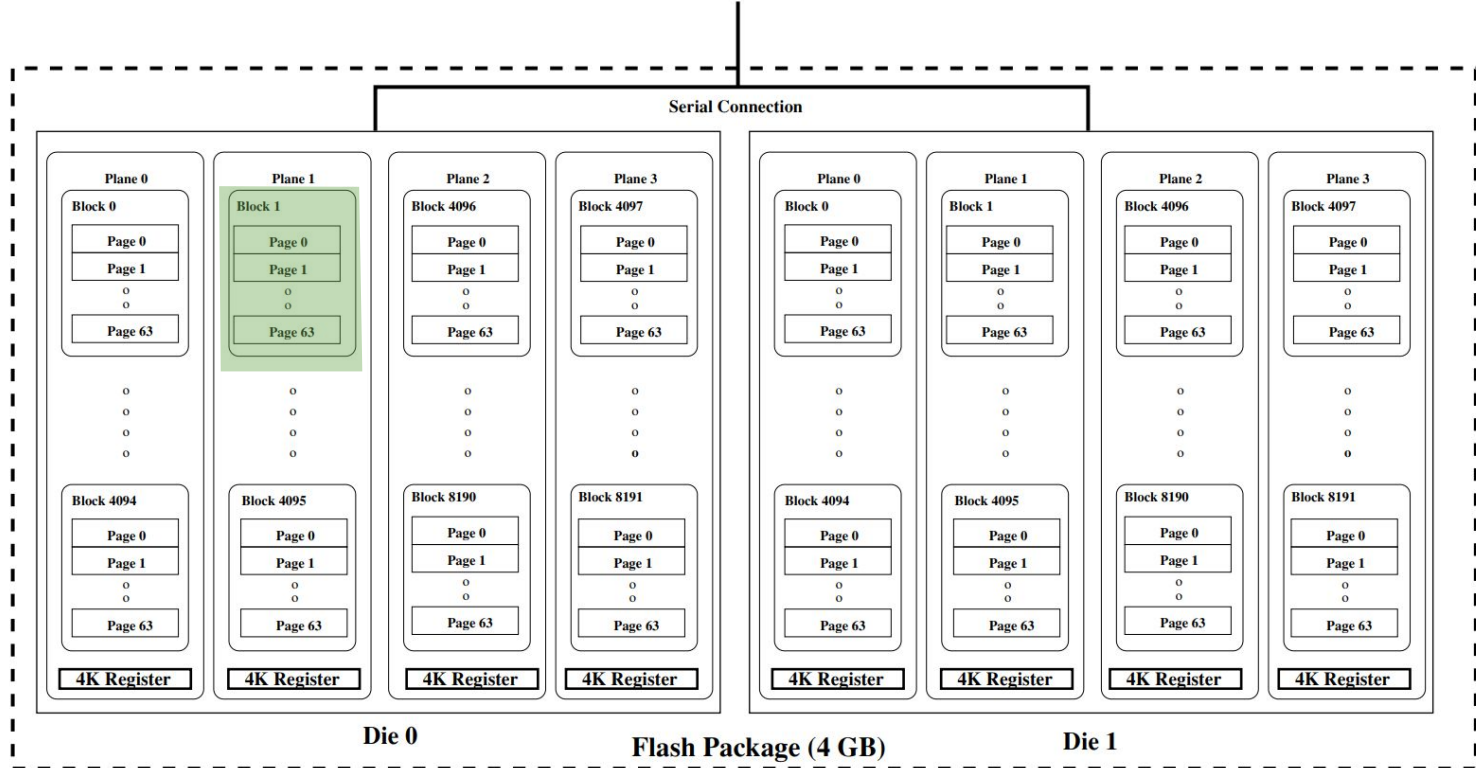
# After



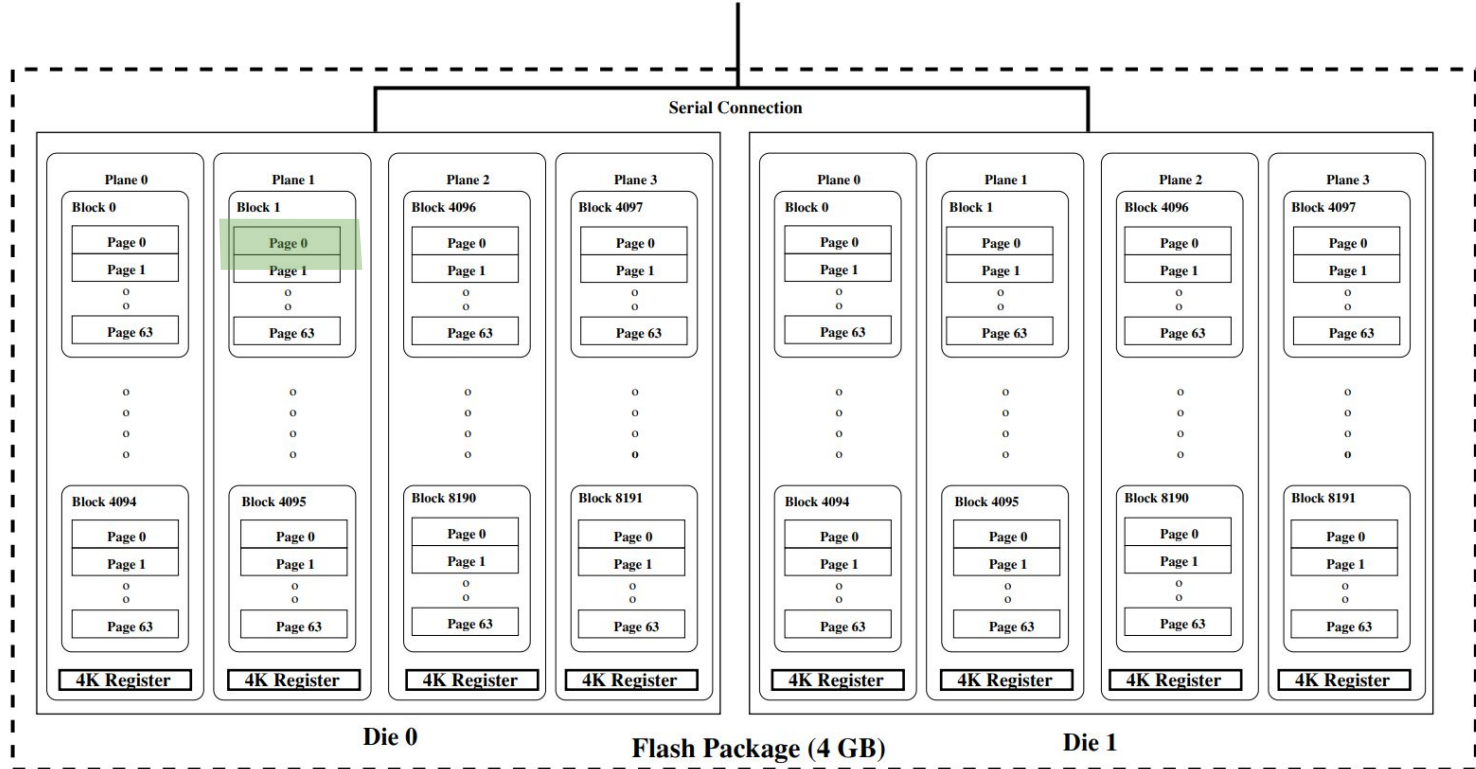
# After



# After



# After



# So, what's new?

<b>HDD</b>	<b>NAND Flash</b>	<b>Optane</b>
Sector granularity R/W	Page granularity R Block granularity E/W	Byte granularity
Slow R/W	Fast R, Slow W	Fast R/W
No Garbage Collection	Garbage Collection	In-place updates
No wearout	Wearout	Wearout



# So, what's new?

<b>HDD</b>	<b>NAND Flash</b>	<b>Optane</b>
Sector granularity R/W	Page granularity R/W Block granularity E	Byte granularity
Slow R/W	Fast R, Slow W	Fast R/W
No Garbage Collection	Garbage Collection	In-place updates
No wearout	Wearout	Wearout

# So, what's new?

<b>HDD</b>	<b>NAND Flash</b>	<b>Optane</b>
Sector granularity R/W	Page granularity R Block granularity E/W	Byte granularity
Slow R/W	Fast R, Slow W	Fast R/W
No Garbage Collection	Garbage Collection	In-place updates
No wearout	Wearout	Wearout

# So, what's new?

<b>HDD</b>	<b>NAND Flash</b>	<b>Optane</b>
Sector granularity R/W	Page granularity R Block granularity E/W	Byte granularity
Slow R/W	Fast R, Slow W	Fast R/W
No Garbage Collection	Garbage Collection	In-place updates
No wearout	Wearout	Wearout

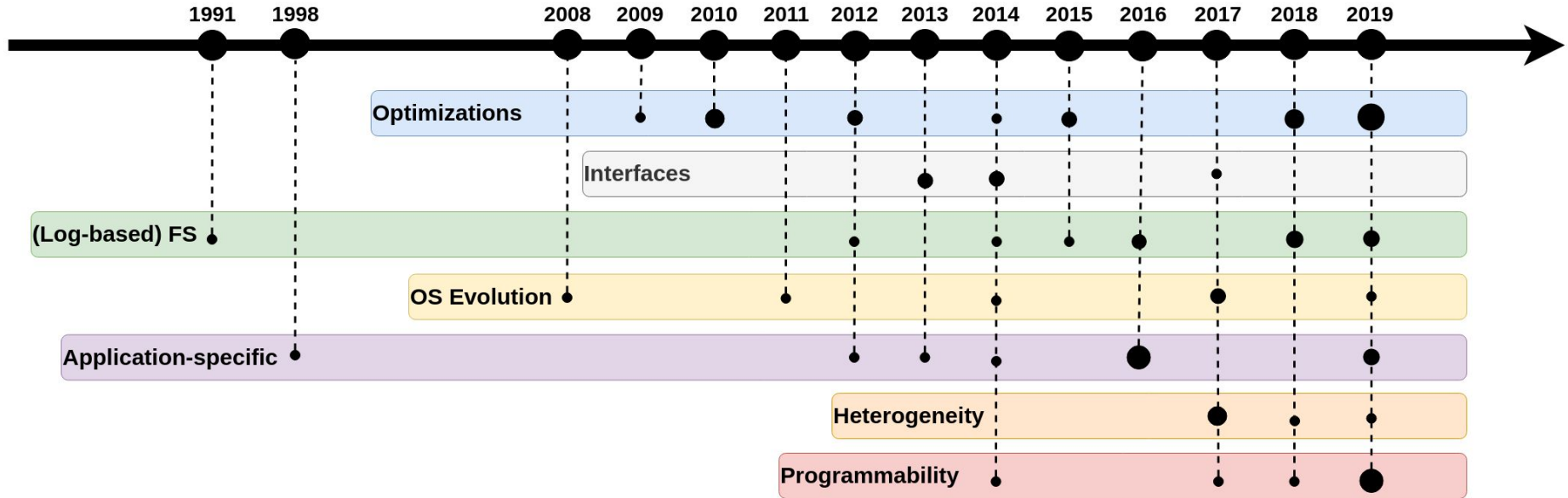
# So, what's new?

<b>HDD</b>	<b>NAND Flash</b>	<b>Optane</b>
Sector granularity R/W	Page granularity R Block granularity E/W	Byte granularity
Slow R/W	Fast R, Slow W	Fast R/W
No Garbage Collection	Garbage Collection	In-place updates
No wearout	Wearout	Wearout

**Where do we even start?**



# The big picture

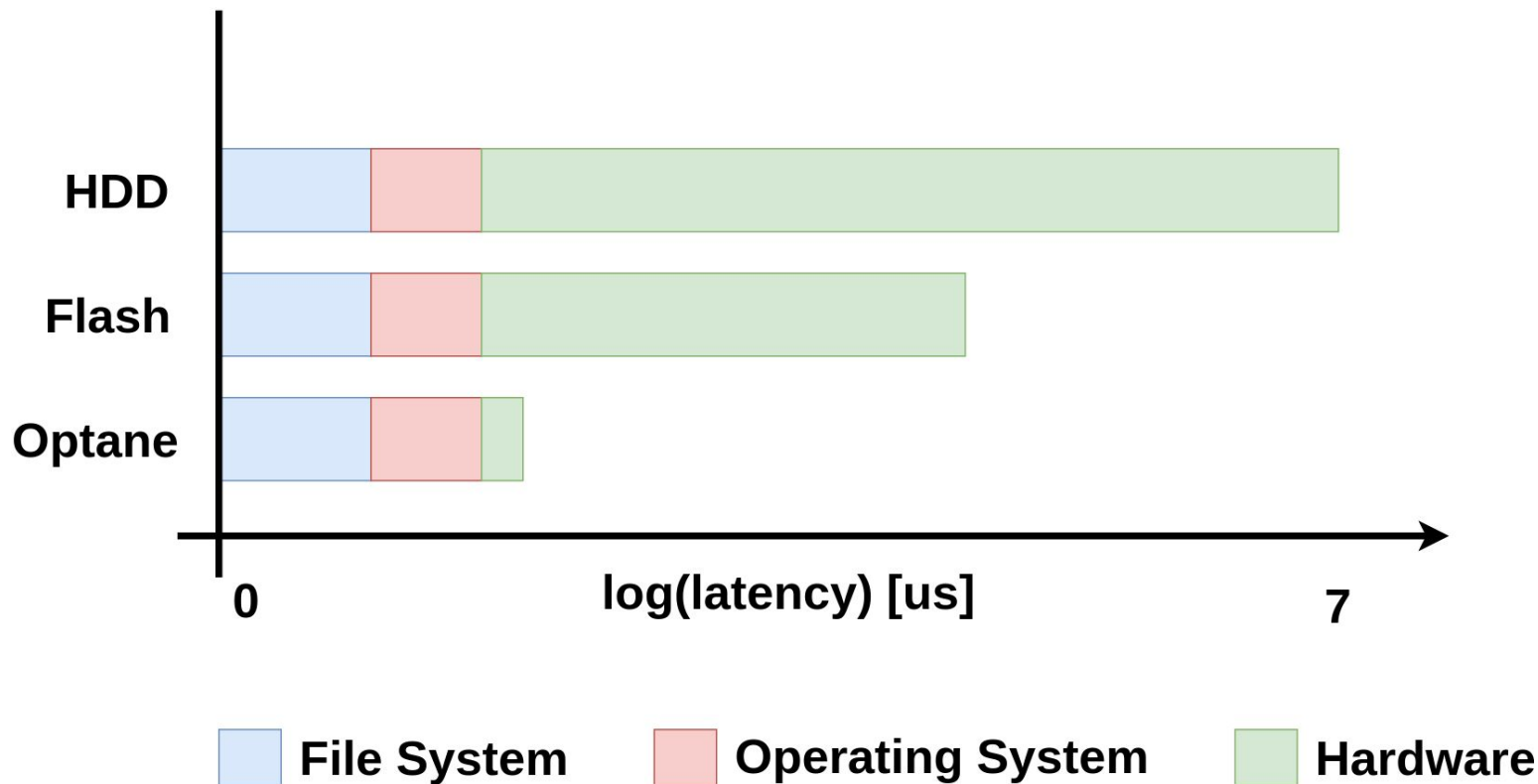


---

# Optimizations & interfaces

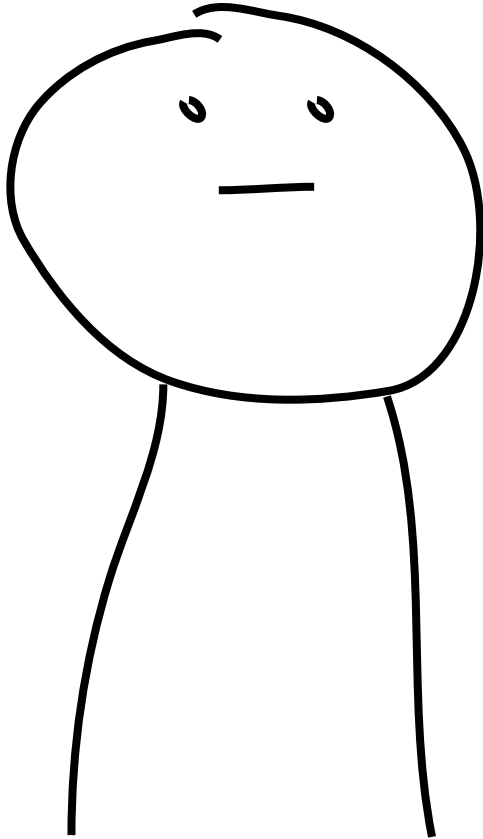
- Revisit low level design decisions
- Old interface based on HDD internals
- Improve bandwidth
- Improve parallelism
- Reduce software overhead

# Latency breakdown

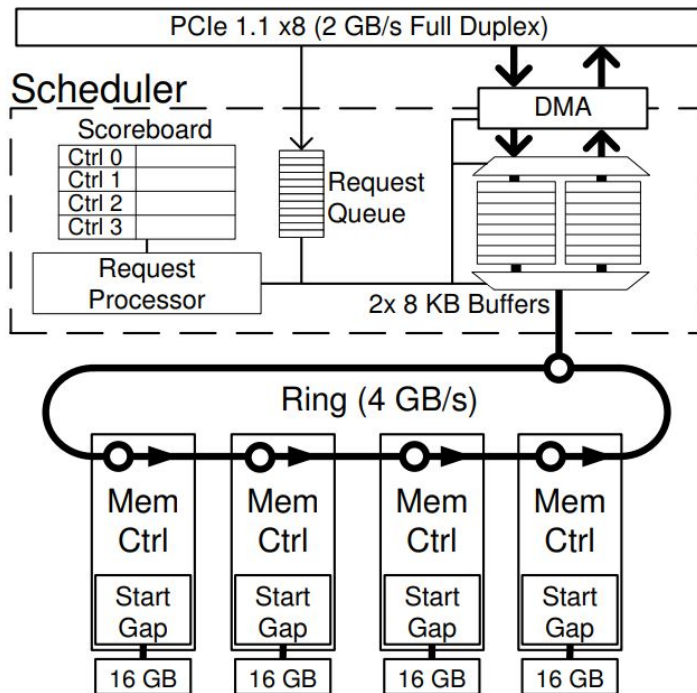




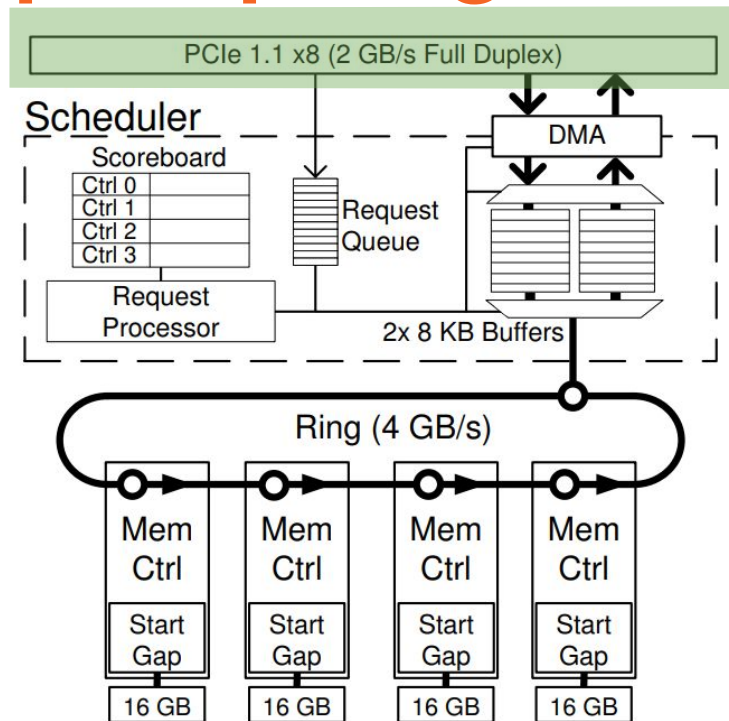
**Now I see...**



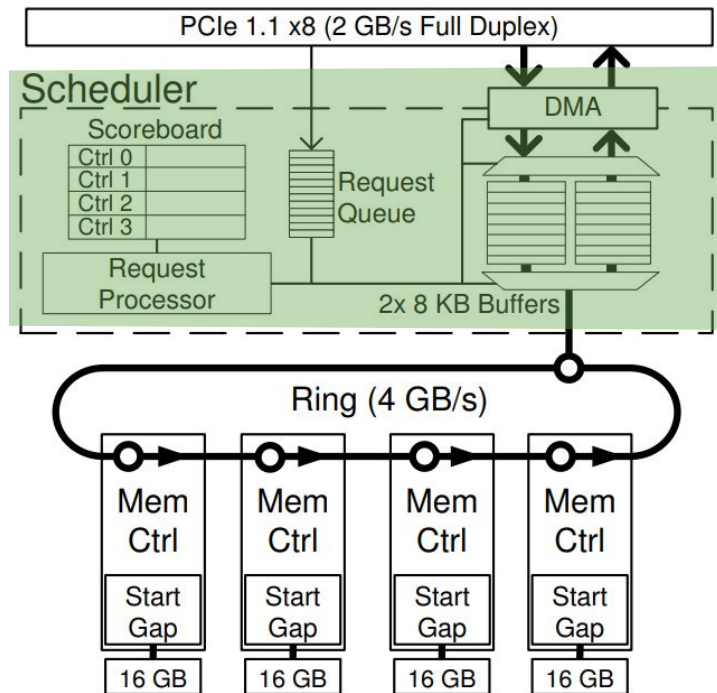
# Paper spotlight: Moneta



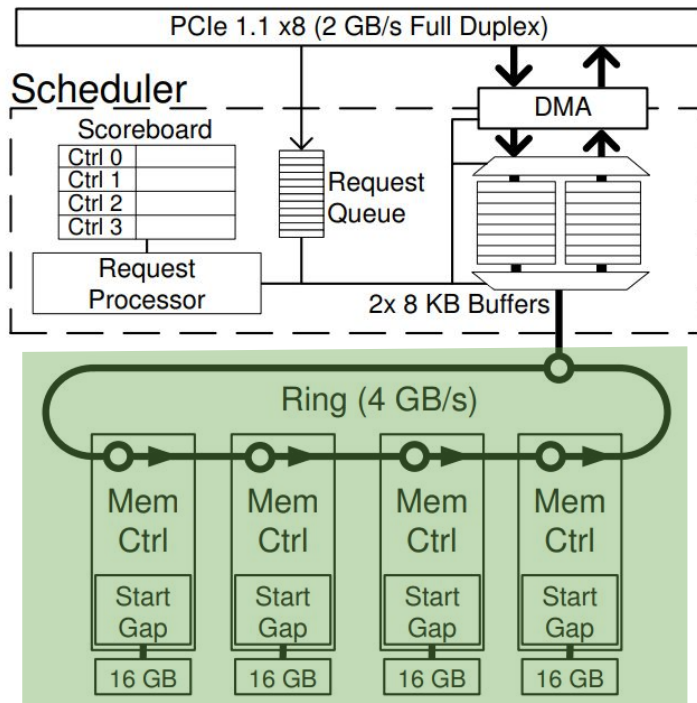
# Paper spotlight: Moneta



# Paper spotlight: Moneta



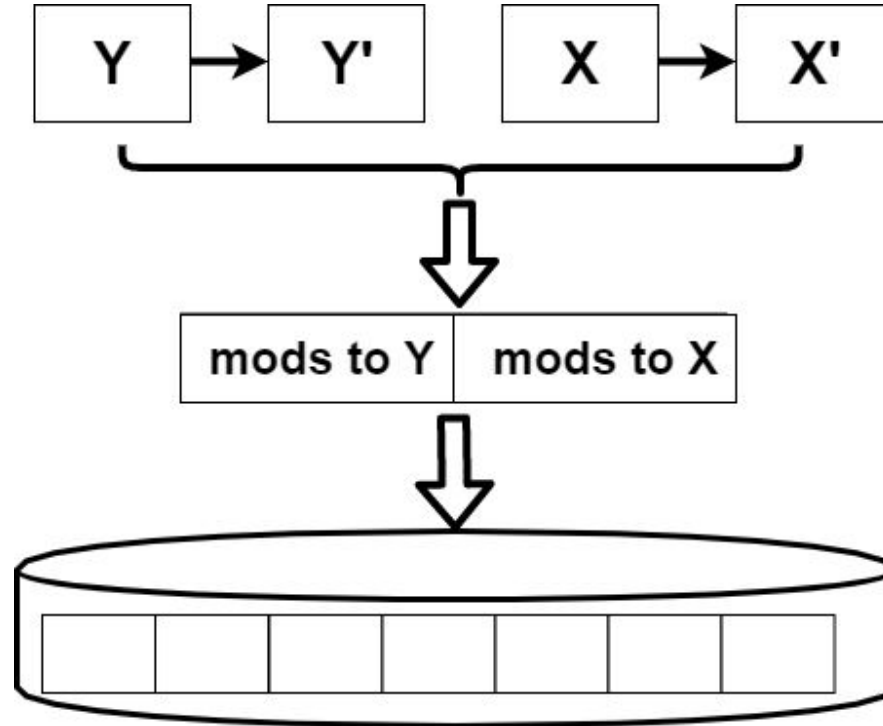
# Paper spotlight: Moneta



---

# What about file systems?

# Log-based file systems



---

# Paper spotlight: F2FS

- Log-based file system
- Mobile and server systems
- 90%+ sequential writes in mobile
- Included in Linux kernel (v3.8)



---

**There's still the OS though**

---

# What the OS has to do

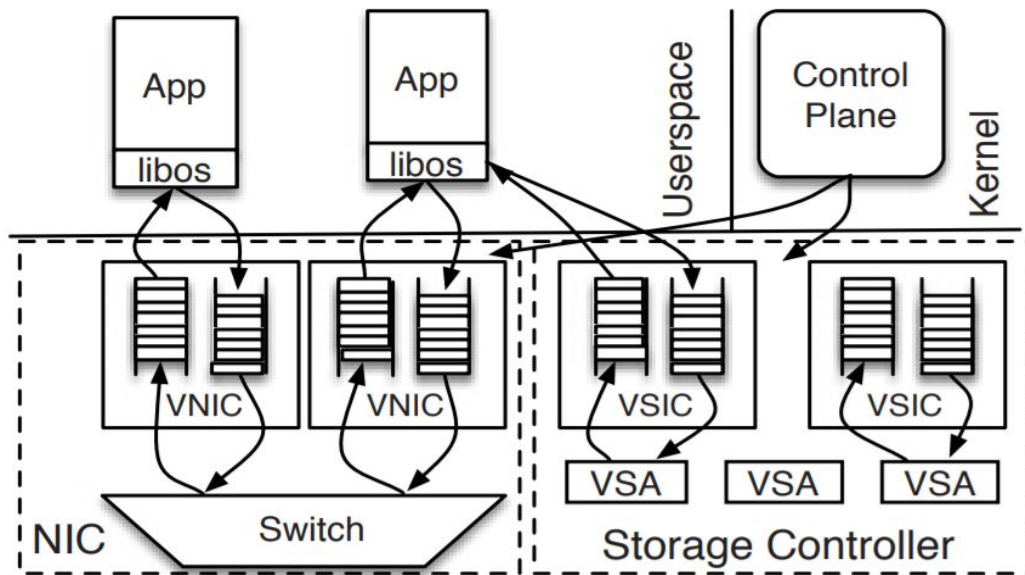
- Multiplexing
- Resource limits
- Protection
- I/O scheduling
- Access control
- Naming

---

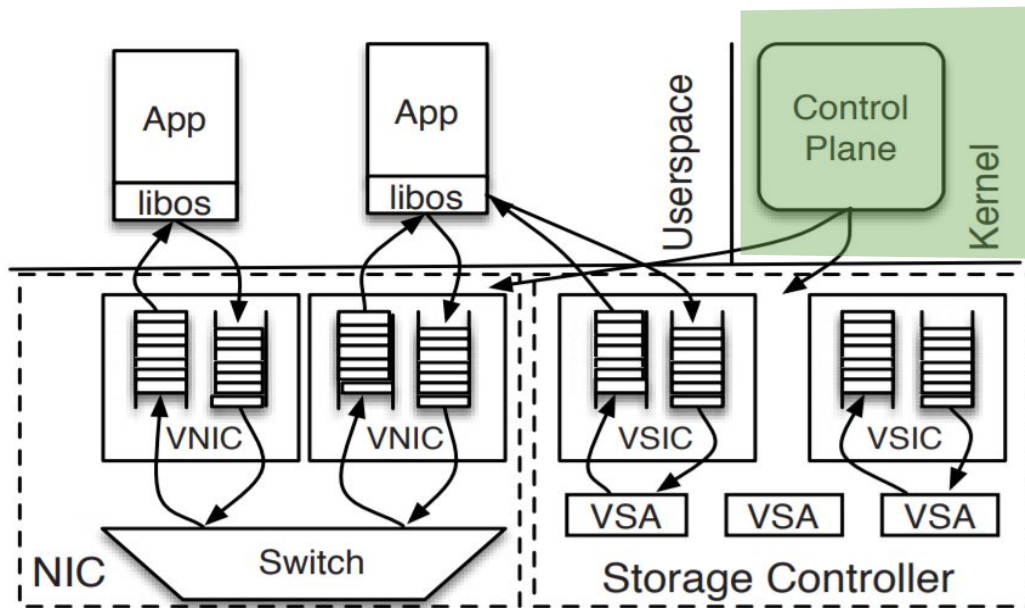
# What hardware can do

- Multiplexing
- Resource limits
- Protection
- I/O scheduling
- Access control
- Naming

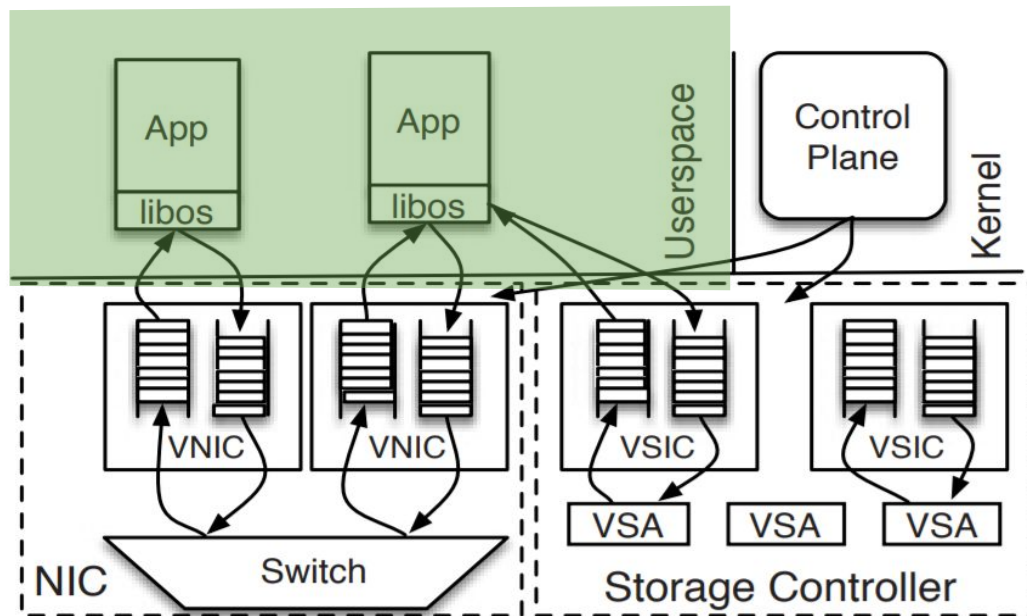
# Paper spotlight: Arrakis



# Paper spotlight: Arrakis



# Paper spotlight: Arrakis



---

# What's next?

---

# Heterogeneity

- New storage is not only an evolution, it is a diversification
- Specialized use cases demand for specialized infrastructure
- DRAM-NVM is not the same as DRAM-HDD



---

## Paper spotlight: NOVA

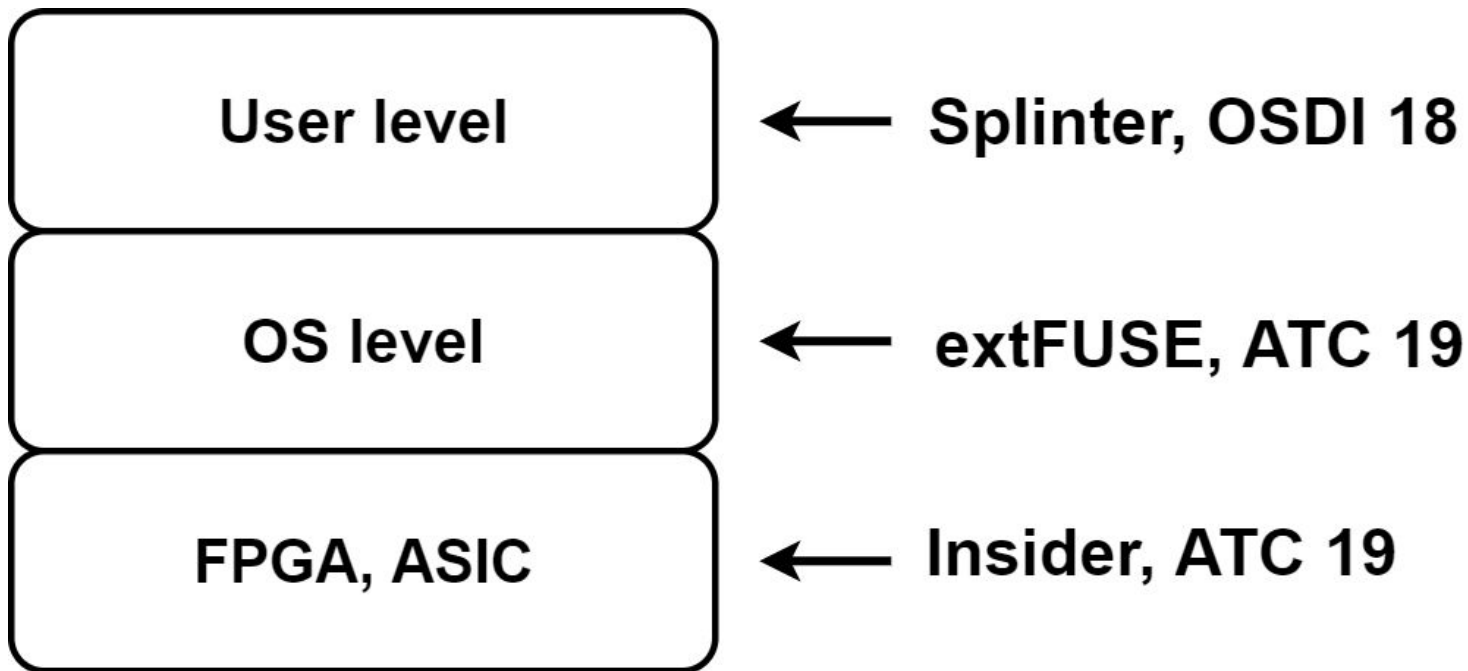
- Traditional file systems do not work with hybrid memory systems
- Log-structured file system
- Hybrid volatile/non-volatile main memory
- Redesign consistency mechanisms

---

# Programmability

- Push custom functionality to the storage
- Want to have multi-tenancy
- Still need to grant safety and isolation
- Can be done at different layers of abstraction

# Programmable everything

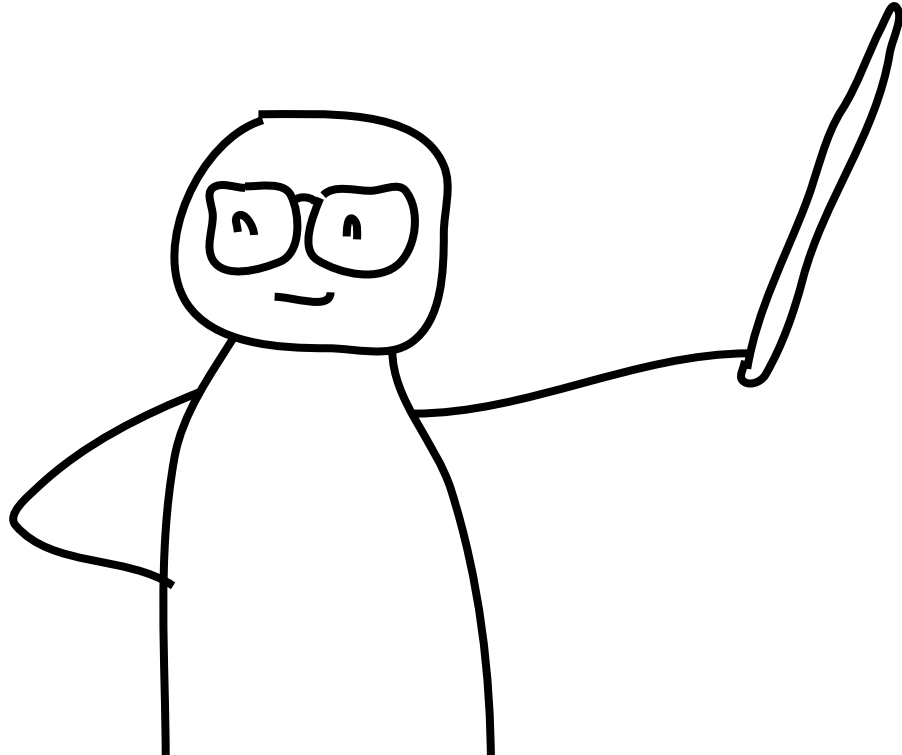


---

# My take

- Programmability is in line with the increase of use cases
- Embrace diversity, don't hide it
- Hardware-based specialization is costly
- Low-level is fun
- Why not do a thesis on this?

**Lesson learned...**  
**There's a lot to research**



---

# With great storage technology comes great responsibility

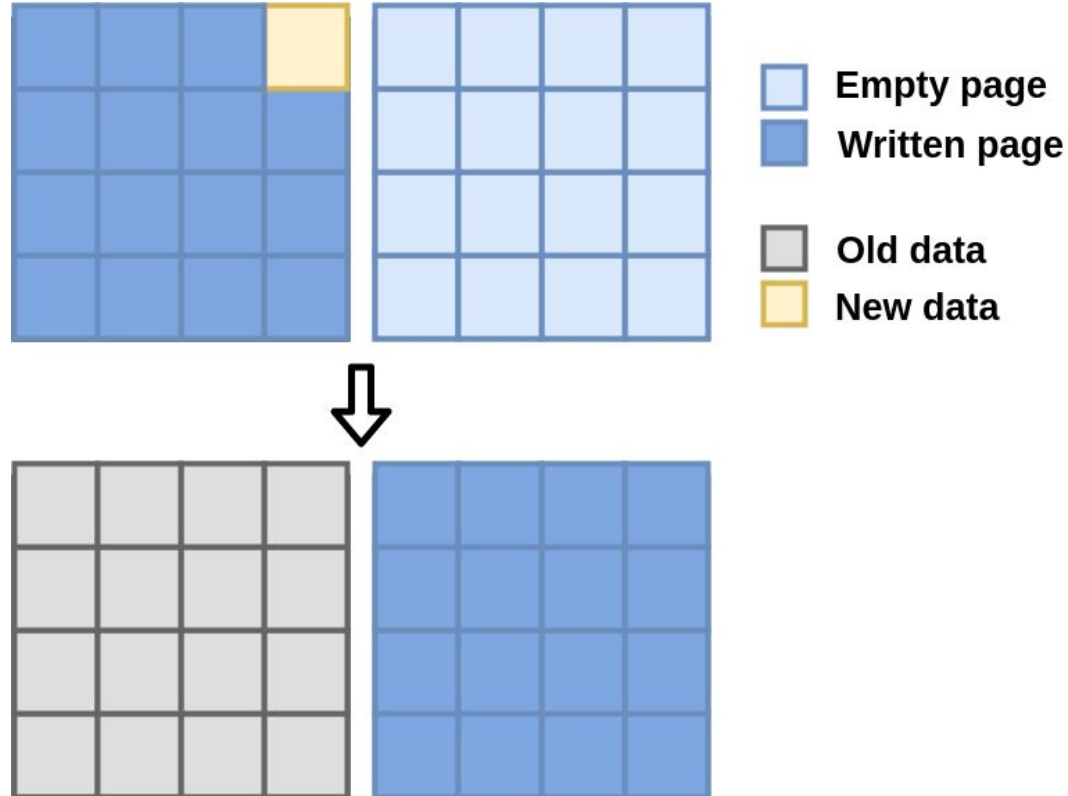
Old solutions won't just “work faster” on new hardware

---

# A note on methodology

- Tier 1 venues (Usenix ATC, FAST, HotStorage, OSDI, SOSP, Systor)
- Focus on the last 5 years
- Include references if relevant (background concepts, fundamental work, vision papers)

# The problem with random writes





# Garbage collection

