

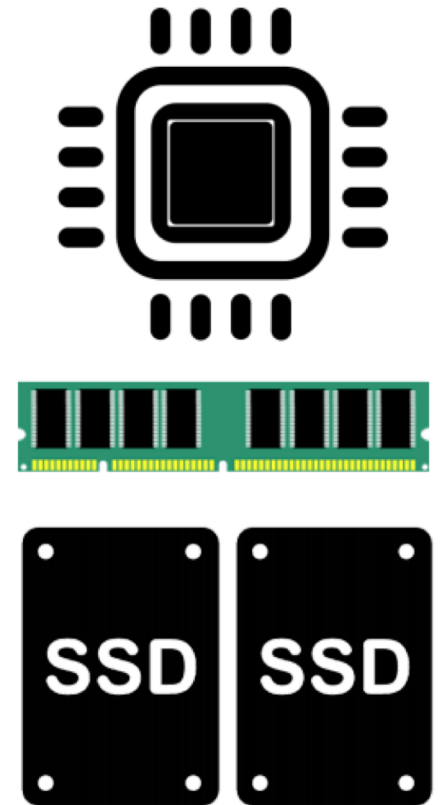
POCS Recitation: Twizzler

Rishabh Iyer

22-10-20

Background: DRAM vs SSDs

- Access latencies?
- What is the granularity of access?
- How do we name objects in each?
- How is an access to each performed?



Enter Non-Volatile Memory

- Access latency?
- What is the granularity of access?
- How do we name objects in each?
- How is an access to each performed?

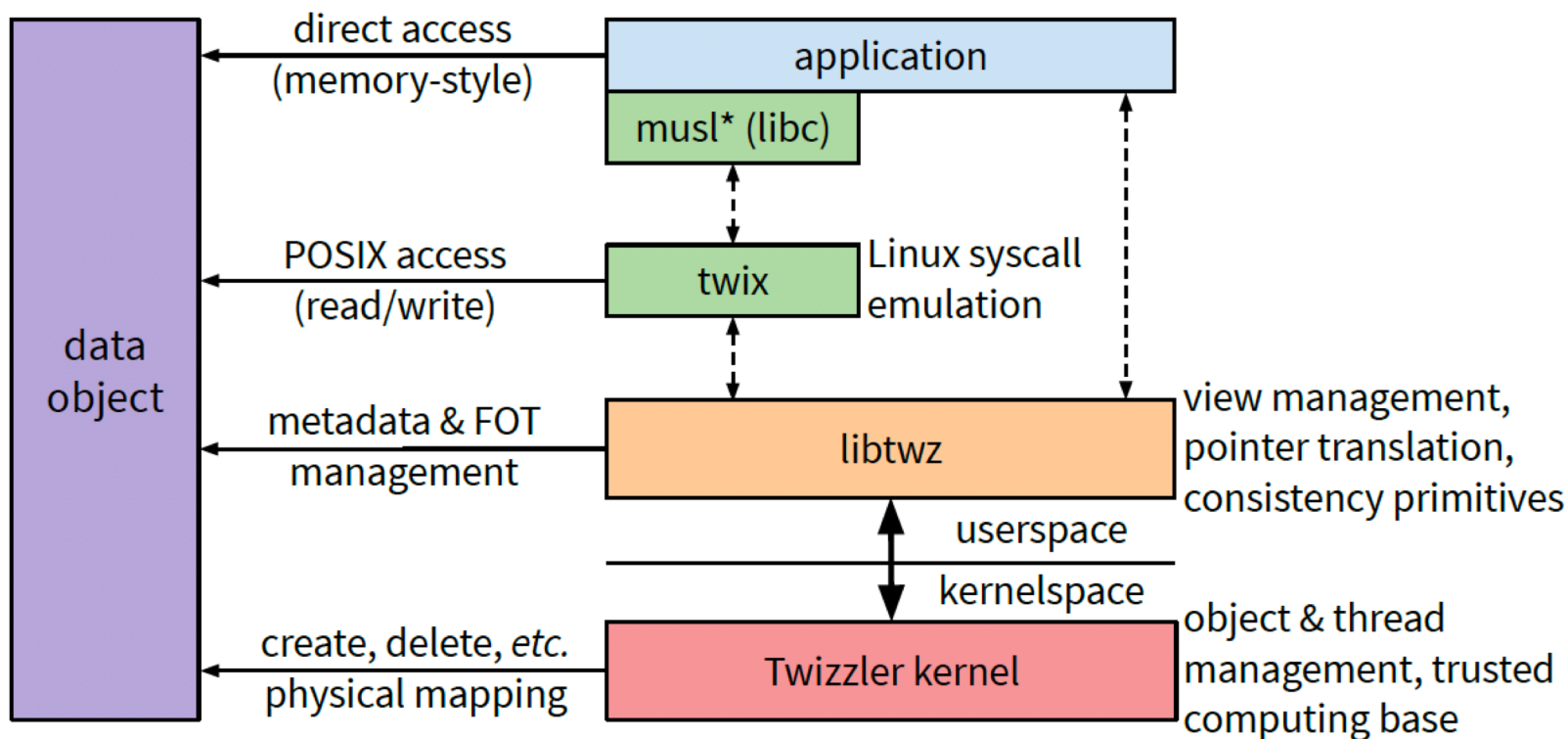
Strawman 1: OS for super fast SSDs

- Access latencies: 2us
- Granularity of access: Block addressable
- How do we name objects in each?
- How is an access to each performed?

The Twizzler OS in an NVM world

- Access latencies: $\sim 500\text{ns}$ to $1.5\mu\text{s}$
- Granularity of access: Byte addressable
- How do we name objects in each?
- How is an access to each performed?

Twizzler Architecture

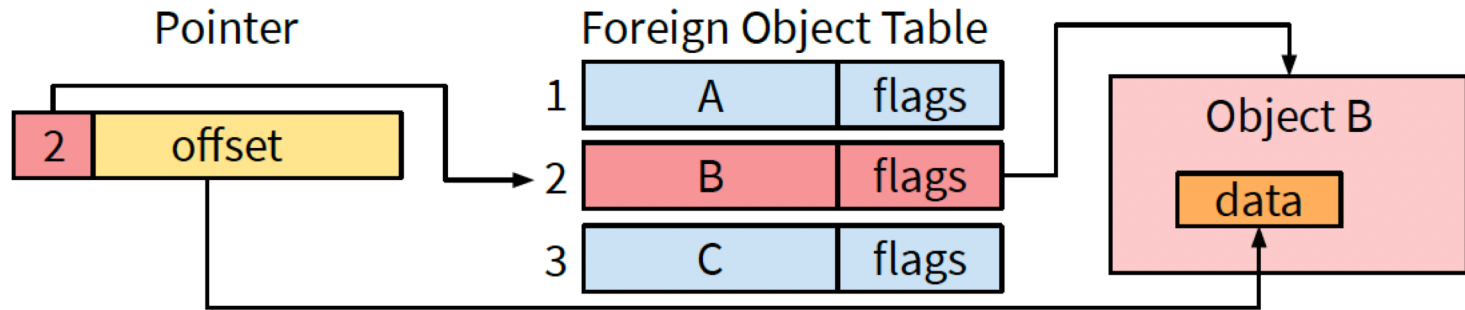


- o Most functionality in libOS (libtwz) to avoid kernel calls
- o Optimise system for tomorrow's software. Provide compatibility for today's software

Persistent Pointers

- Why are they needed?
 - How do I share a linked-list across machines?
- How does Twizzler implement persistent pointers?

Foreign Object Table



- Why add a level of indirection?
- How are pointers translated?

Summary

- Today's storage is block-addressable and slow
- So we're using the POSIX standard and are happy with an in-kernel FS.
- Twizzler: first OS that treats persistent objects as first-class citizens.
 - This requires the design and implementation of persistent pointers