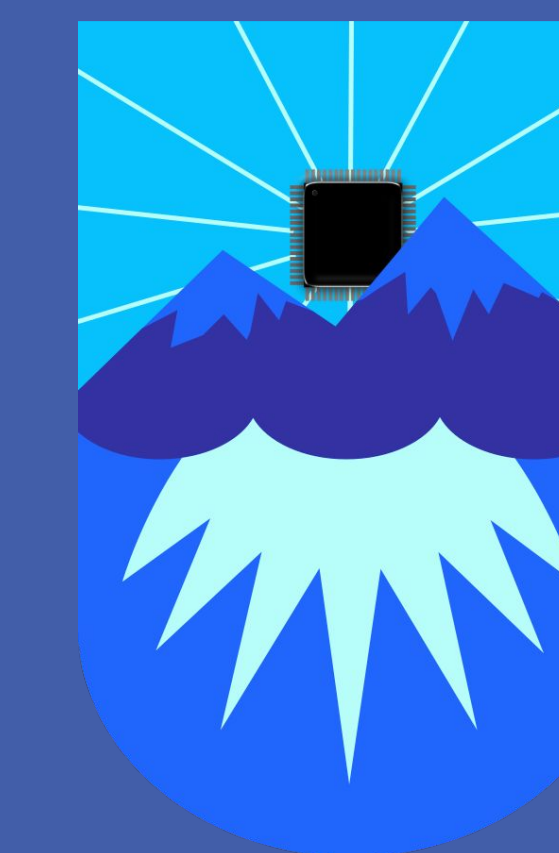




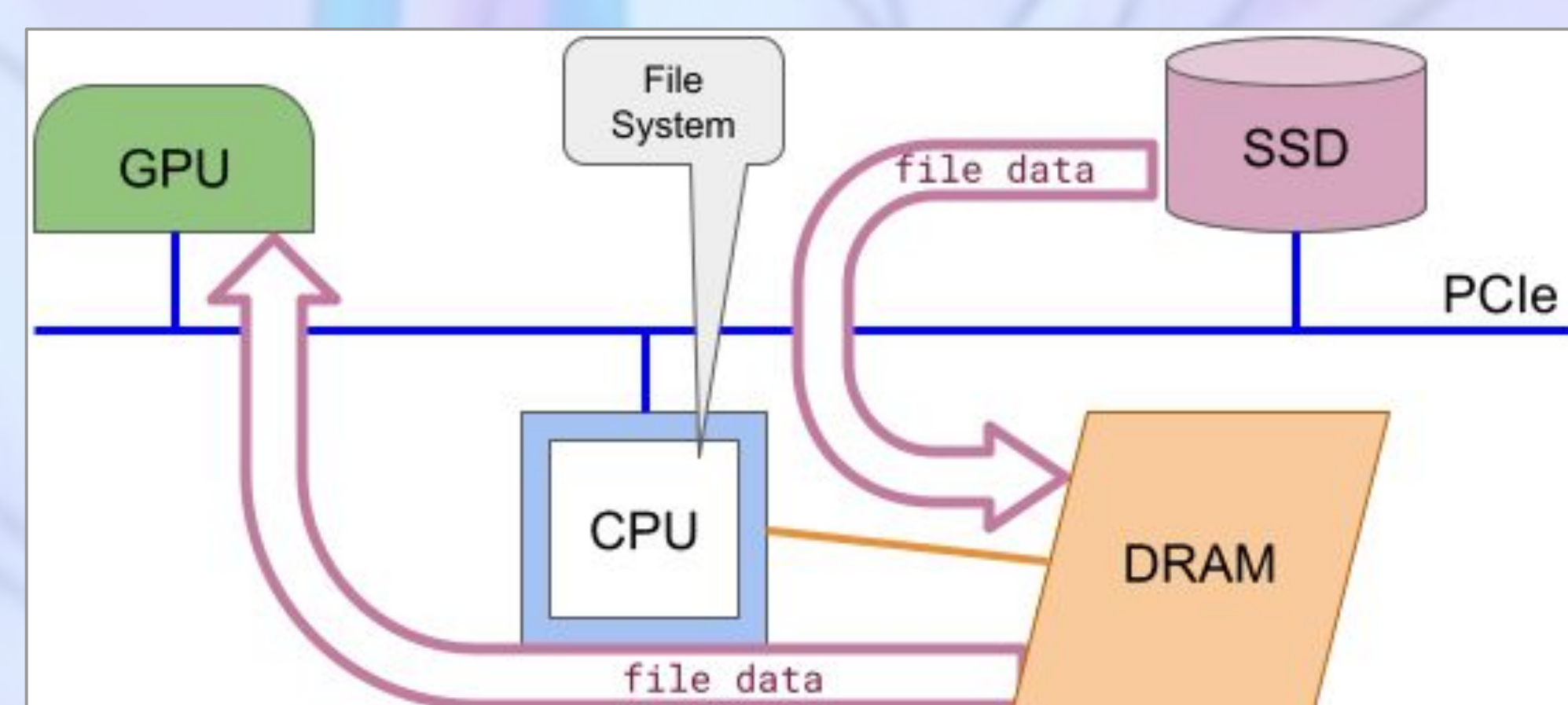
# Offloading File Systems from the CPU

Joel Nider, Sasha Fedorova  
Department of Electrical and Computer Engineering  
University of British Columbia



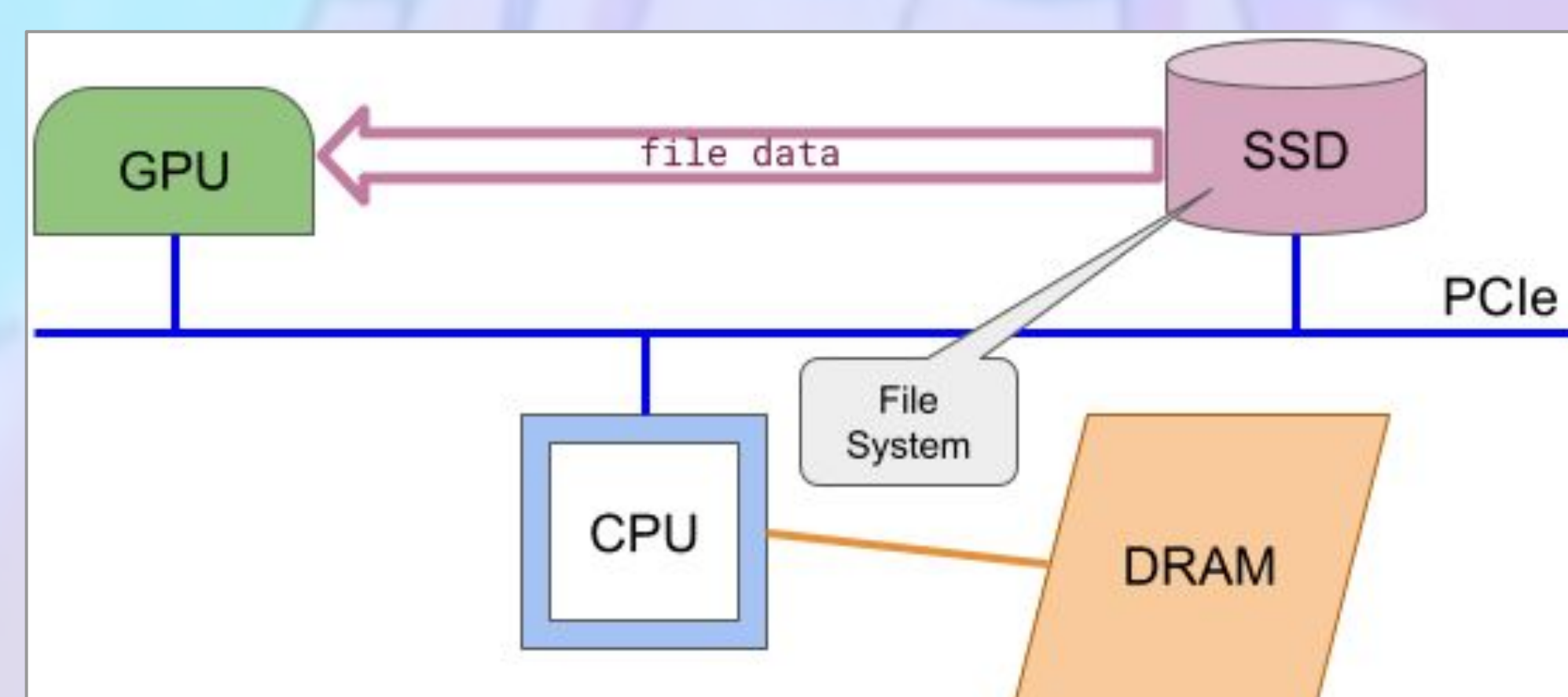
## Problem

Significant CPU time is spent on file system processing. Especially when the files are used by an accelerator (e.g. GPU) this increases latency, complexity and cost of the overall system.



## Solution

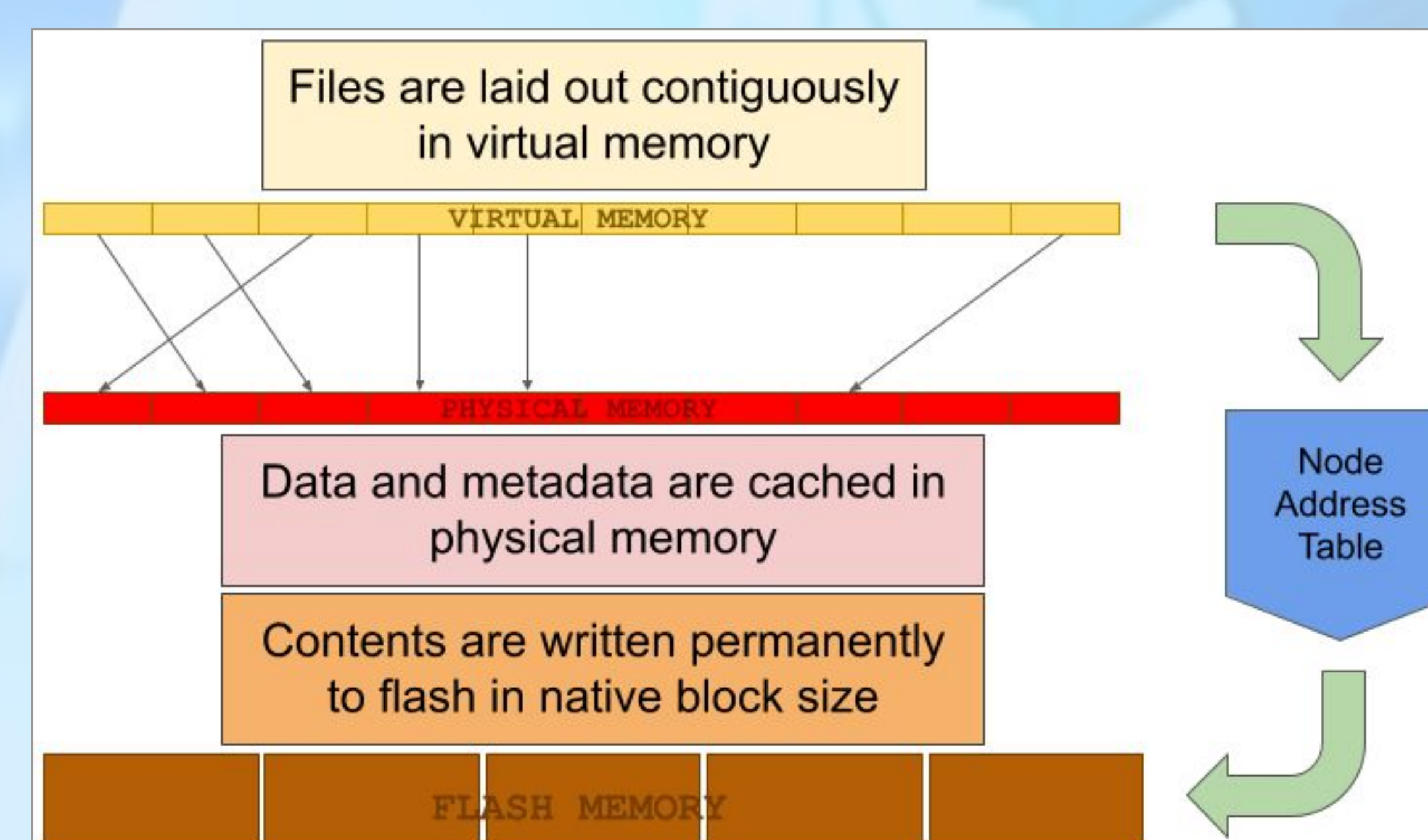
Offloading the file system to the SSD frees up precious CPU cycles on the host and localizes I/O processing.



## Design

The file system is designed for flash memory devices (i.e. SSD). The key features are:

- Uses a familiar file system interface
- Eliminates redundancies between the file system and flash translation layer
- Eliminates garbage collection
- Exposes files to all peripherals & CPUs
- Uses virtual memory inside the drive to eliminate indexing overhead



We can take advantage of existing drivers in Linux because the drive uses *virtio-fs* (a combination of the **virtio** protocol and **FUSE** command set).

Data remains in the drive's cache as long as possible. Changes are written to flash periodically or on-demand.

## Benefits

- Reduces host CPU load
- Reduces data movement
- Any device with access to the memory bus can access the file system
- Reduces overhead imposed by separate file system and FTL layers
  - file system block granularity is dictated by flash erase block size

## Future

- User space library implementation (no system calls)
- A single file system across multiple hardware devices (e.g. RAID)