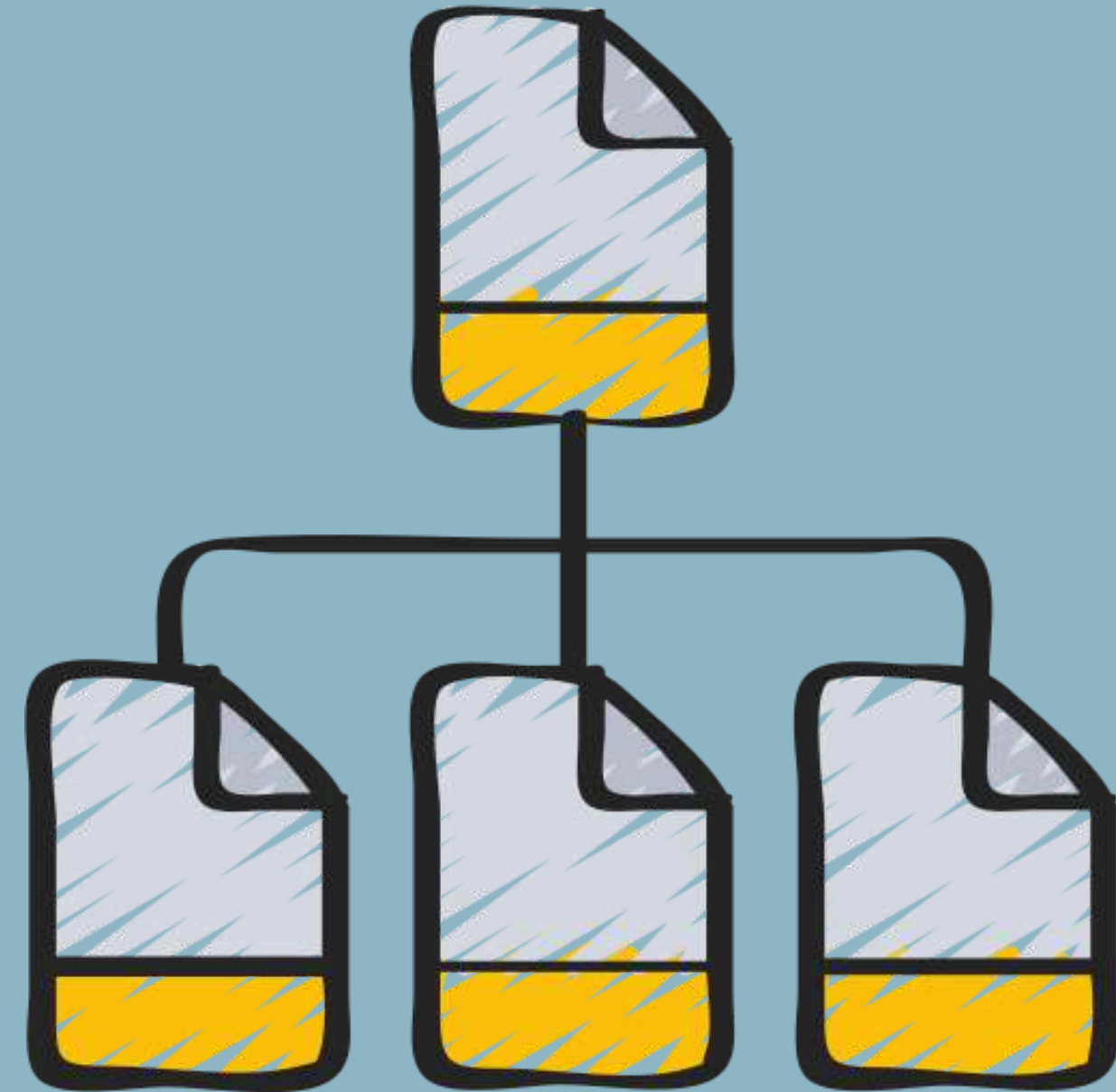


16 décembre 2024

UFS

Un système de
fichiers semi-
microkernel



01

Origine

- Projet de recherche à **Université de Wisconsin-Madison**
- Chef de recherche : **Jing Liu**
- Embauche chez **Microsoft Research Asia**



WISCONSIN
UNIVERSITY OF WISCONSIN-MADISON

Microsoft[®]

Research

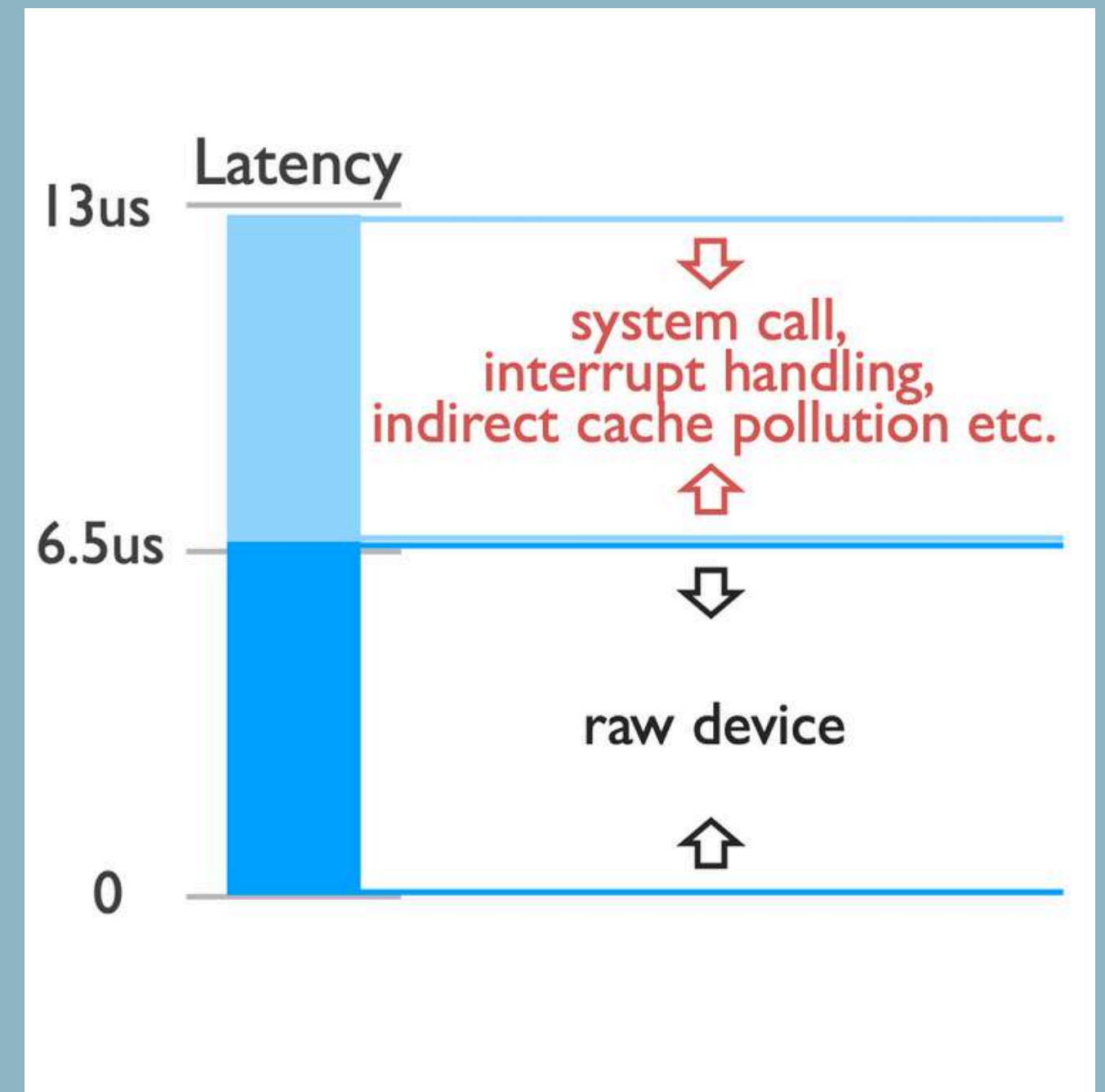
微软亚洲研究院

02

Contexte

Les Dispositives Modernes

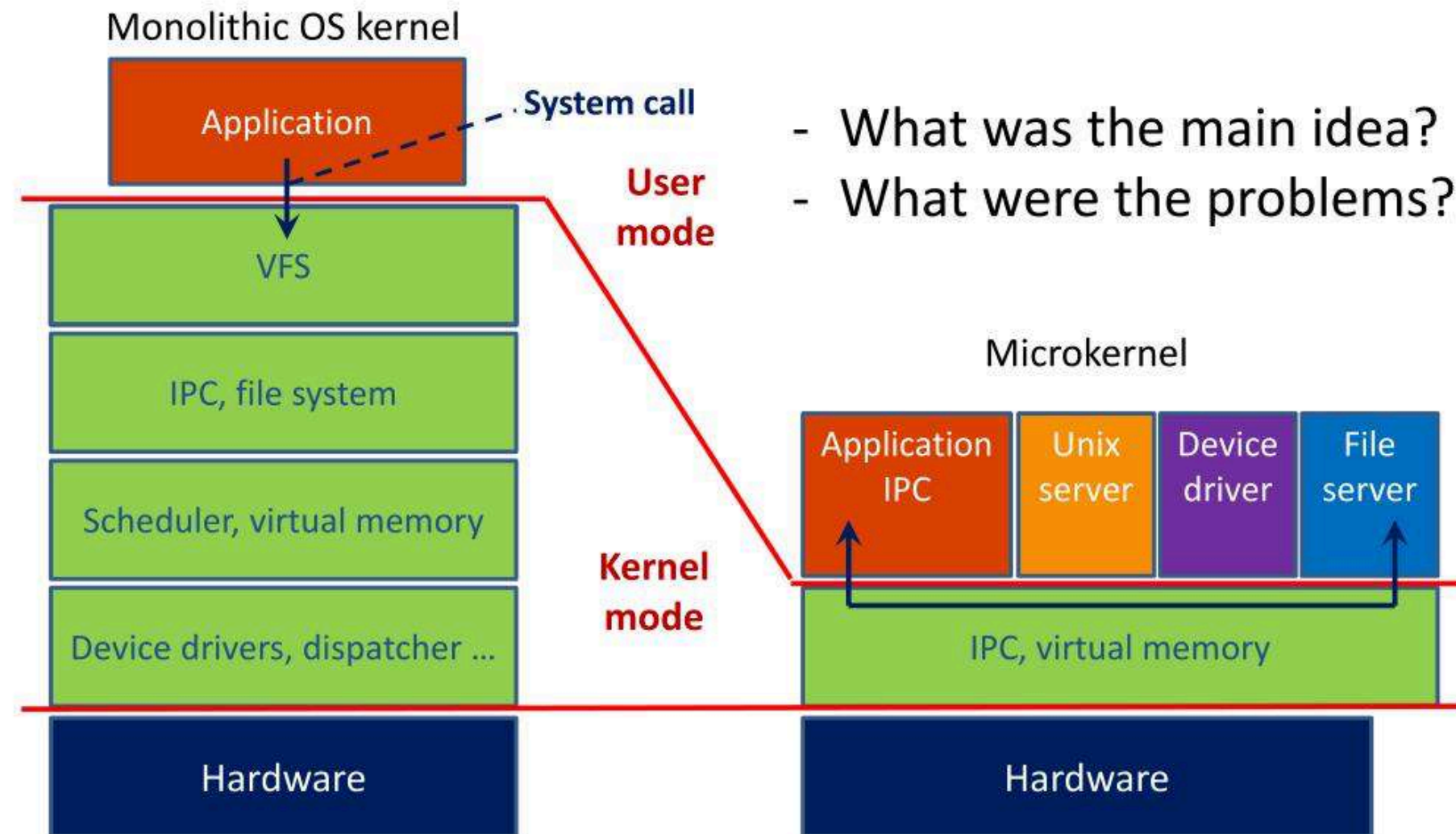
- **Hardware est rapide, mais le software est long**
- **SSD NVMe : latence faible que 6.5 microseconde**
- Le noyau traditionnel **double** cette latence

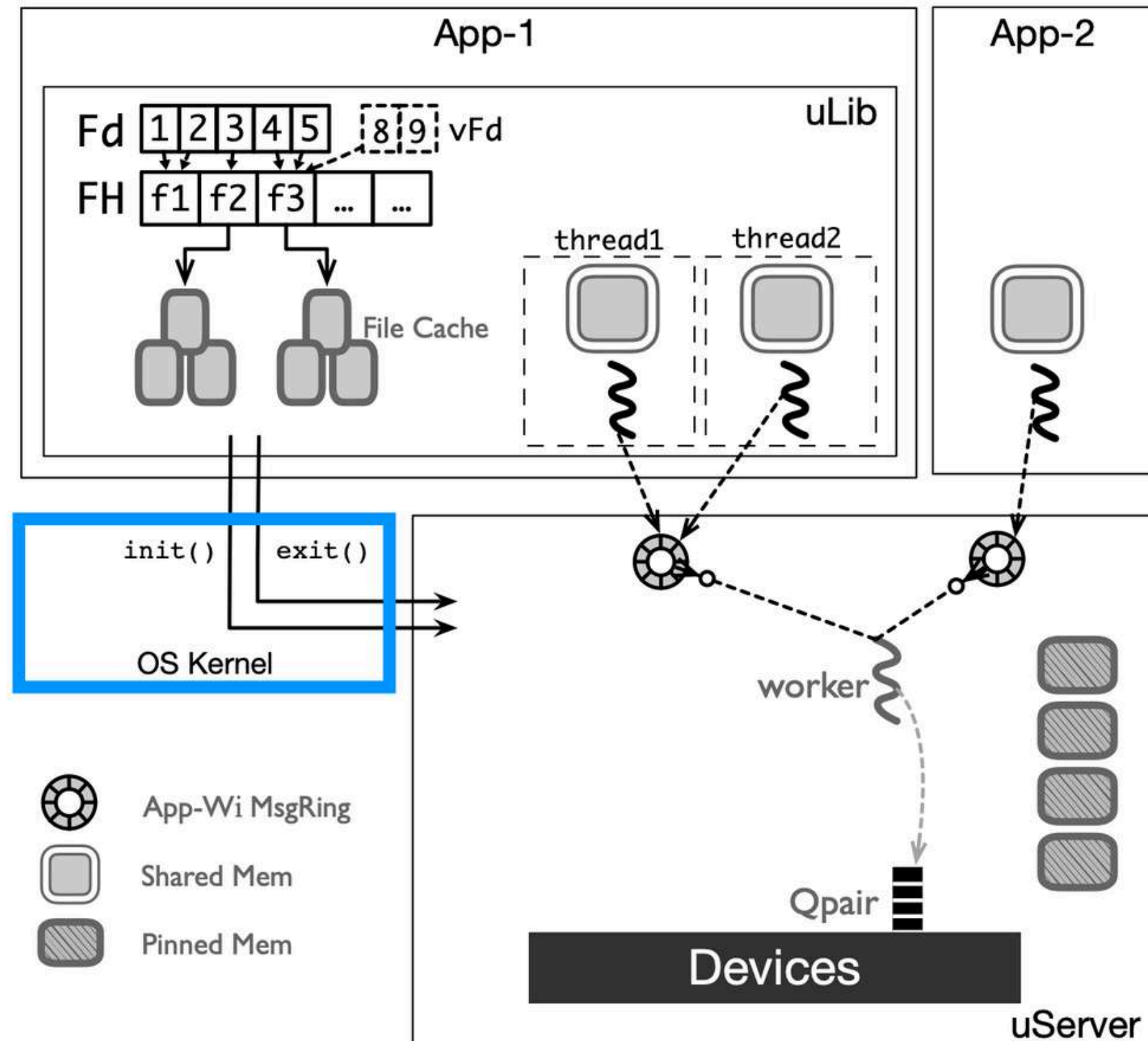


Terminologies

- **IPC** : Interprocess communication
- **VFS**: Virtual FileSystem.

Monolithic kernel vs Microkernel





Les composants

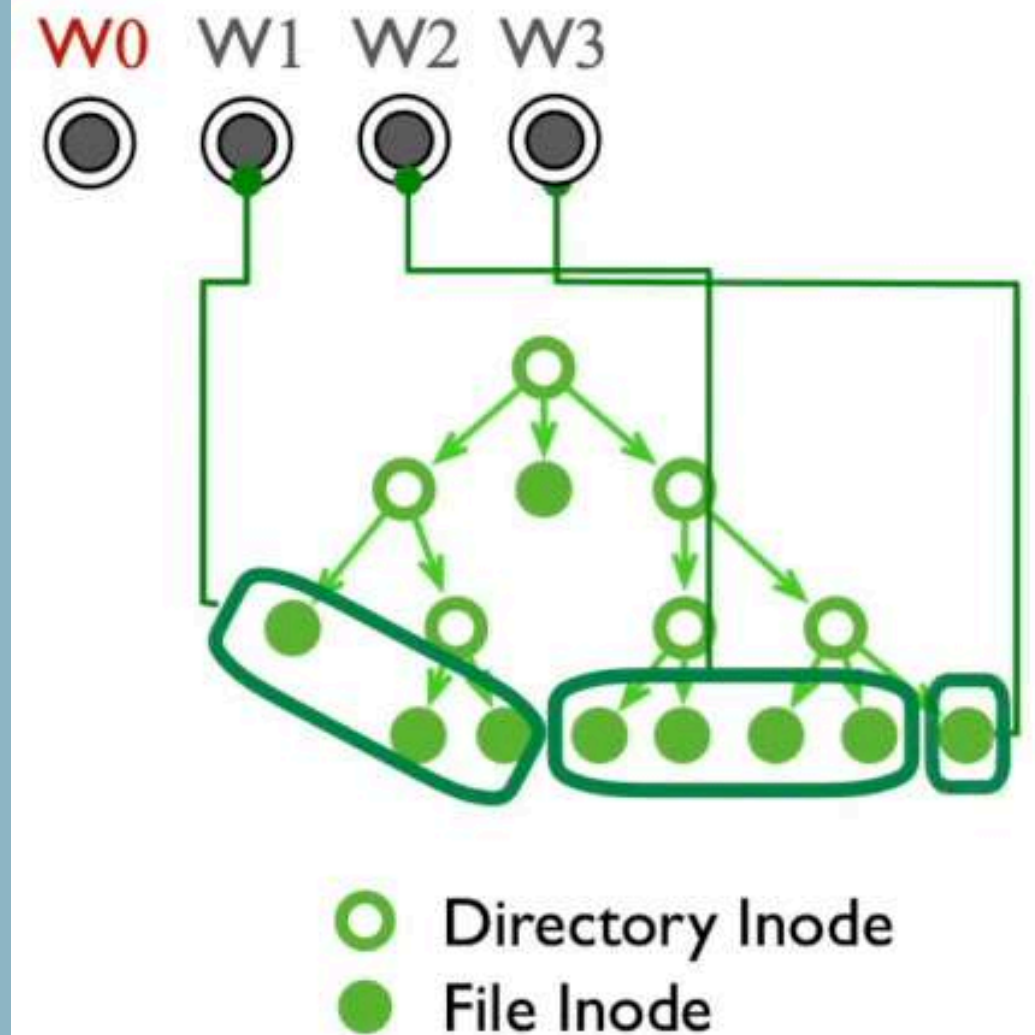
- Une Architecture **semi-microkernel**.
- **uLib**
 - API fournit aux applications
 - peut être appelé par les différentes applications.
- **uServer**
 - Permet la communication directe avec les appareils de stockage.
 -
- **OS Kernel**
 - fait des opérations basiques.
 - **init()** et **exit()** seulement.

05

Mécanismes Clés

Runtime Inode Ownership

- Chaque groupe d'inodes est **accessible exclusivement** par un **worker**
- **Worker asymétrique**
 - Un worker principale (W0)
 - Des workers secondaires : file ops
- **Decouplage** des namespace et du ownership
 - permet l'obtention d'une inodes **par 2 workers**

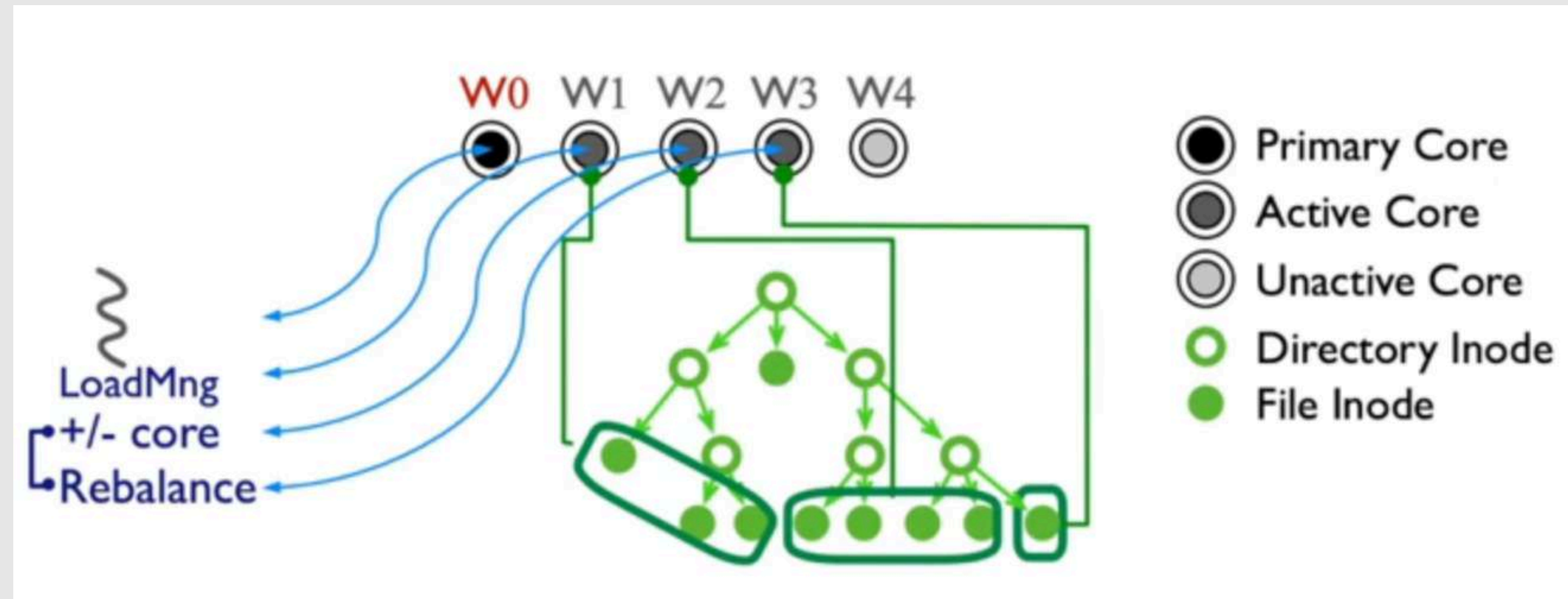


06

Gestion de la Charge

Les composants

- **Thread** dédié pour le **Load Balancing**
- **Collecte périodique** des données de chaque worker
- Décide de l'**objectif de charge** pour chaque worker ("What if")
- **Réassignement** de d'inodes pour l'équilibrage

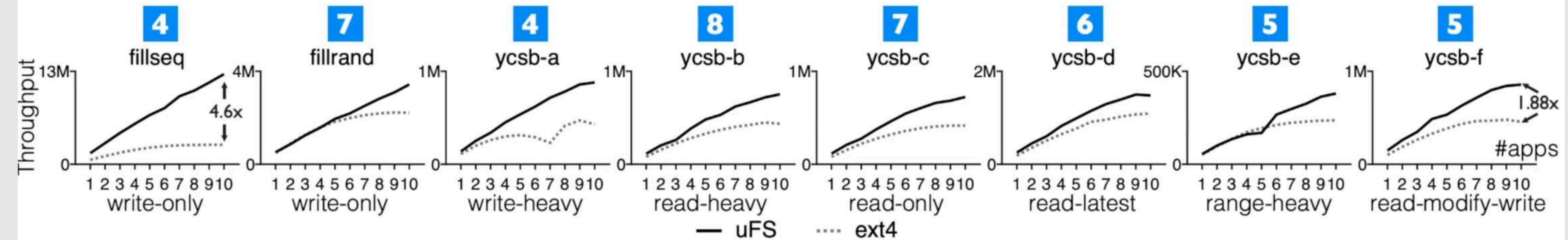


07

Performance

Comparaison

Number of cores (when #app=10)



uFS évolue beaucoup mieux que **ext4**

uFS allouera un nombre différent de cœurs pour différentes **charges de travail**.

NB : Donner plus de cœurs (> 10) à d'ext4 n'aide pas beaucoup à la performance

Avantages	Inconvénients
<p data-bbox="793 1009 1436 1065">Haute performance</p> <p data-bbox="942 1174 1286 1230">Scalabilité</p> <p data-bbox="696 1339 1512 1395">Adaptabilité aux charges</p> <p data-bbox="739 1504 1469 1560">Développement rapide</p>	<p data-bbox="1805 926 2635 983">Complexité de conception</p> <p data-bbox="1829 1091 2612 1148">Dépendance au matériel</p> <p data-bbox="1769 1256 2675 1313">Nécessite des optimisations</p> <p data-bbox="1745 1422 2698 1478">Maintenance potentiellement coûteuse</p>

uFS

Version **Experimental** License **MIT**

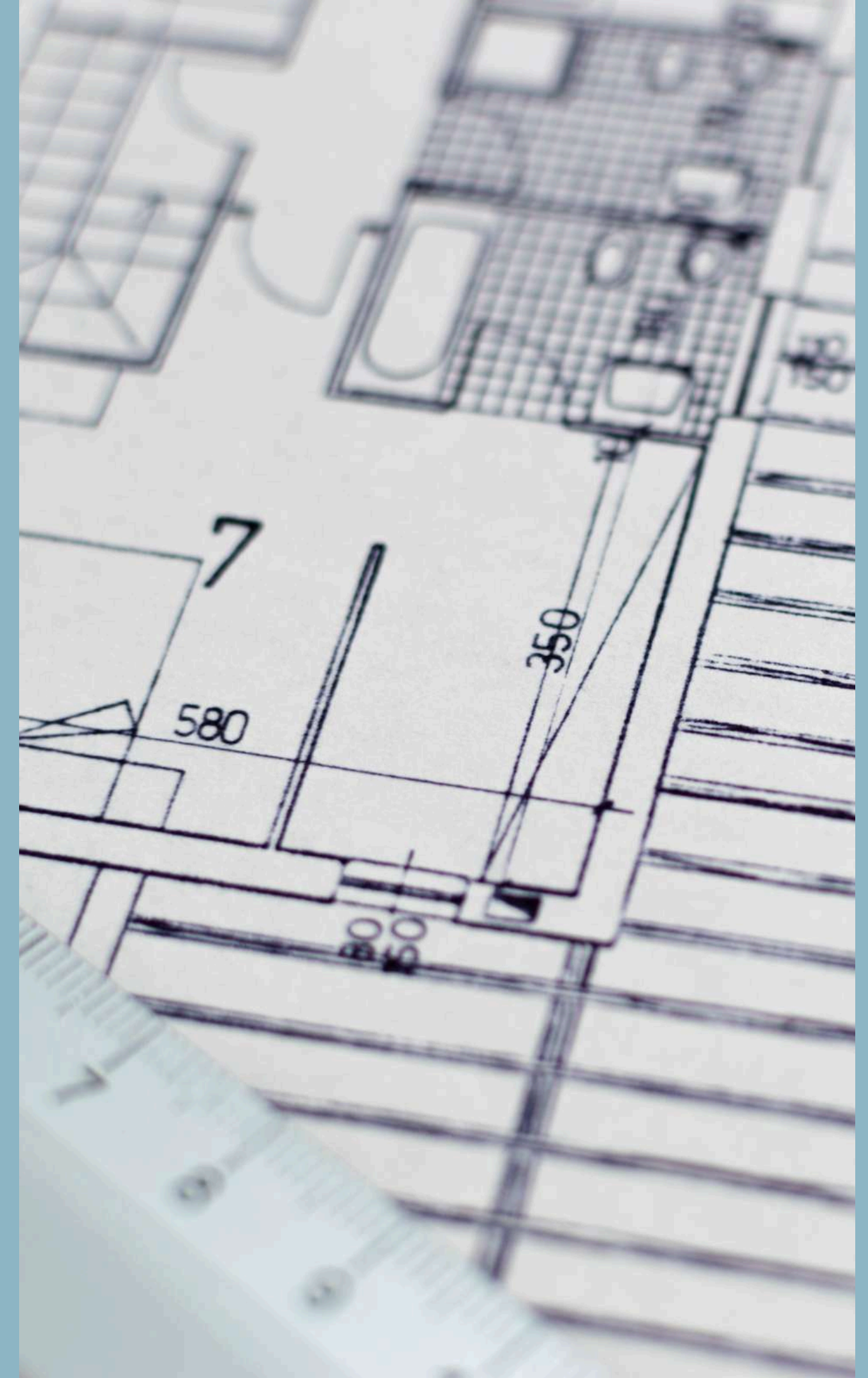
uFS is a filesystem semi-microkernel that designs for device performance delivery and scalability. We primarily design for modern ultra-fast NVMe devices and multi-core machine. uFS runs as a standalone user-level filesystem process, provides POSIX compatible APIs and crash consistency guarantee, and dynamically adapts to the application demands.

You could learn more about uFS design and semi-microkernel approach in our SOSP'21 paper [Scale and Performance in a Filesystem Semi-microkernel](#).

Repo Structure

```
uFS
|---- cfs
|---- include          # uFS's cpp headers
|---- lib              # uFS's dependent libraries
|---- src              # uFS's cpp source code
|---- test             # tests for uFS, including utility tools like cli and mkfs
|---- tools
|---- cfs_bench
|---- bench            # cpp source code for uFS's microbench
|---- exprs            # scripts to run uFS's experiments
|---- helpers
|---- include
|---- port
|---- util
```

The other benchmarks in our paper besides microbenchmark (`cfs_bench`) are in a separate repo [uFS-bench](#), including `filebench`, `leveldb` and `scalefs_bench`.



uFS : Un système de fichiers semi-microkernel

- Conçu pour le **stockage modernes** performants et **évolutifs**
- **Plus performant que ext4** pour les charges de travail
- Évolue indépendamment des applications et **s'adapte dynamiquement à la demande**

Approche Semi-Microkernel

- Performant et évolutif sous différentes charges de travail
- Tous les avantages d'un développement au niveau de l'utilisateur

- ACM SIGOPS. (2021, 21 octobre). SOSP 2021 : Scale and Performance in a Filesystem Semi-Microkernel [Vidéo]. YouTube.
<https://www.youtube.com/watch?v=J7Ou48NKUis>
- Liu, J., Rebello, A., Dai, Y., Ye, C., Kannan, S., Arpaci-Dusseau, A. C., & Arpaci-Dusseau, R. H. (2021). Scale and Performance in a Filesystem Semi-Microkernel. University Of Wisconsin-Madison, 819-835.
<https://doi.org/10.1145/3477132.3483581>
- WiscADSL. (s. d.). GitHub - WISCADSL/UFS : a filesystem Semi-Microkernel. GitHub. <https://github.com/WiscADSL/uFS>

Merci pour votre attention !

Avez-vous des questions ?