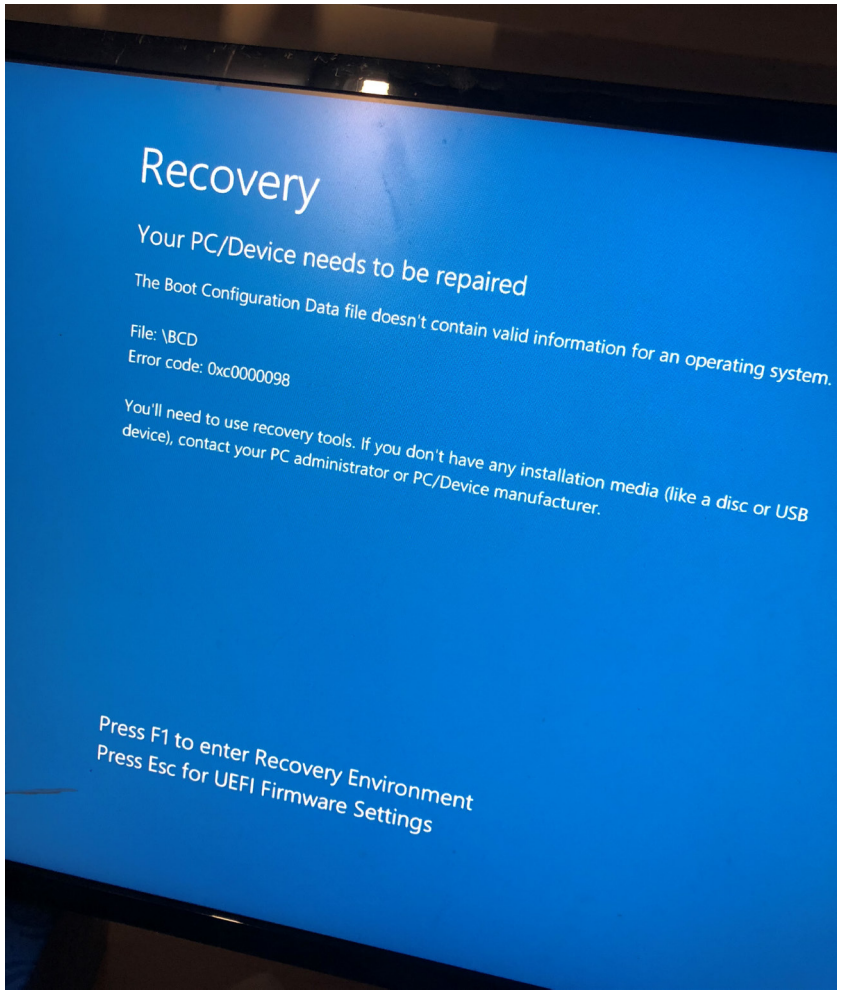
APRON: Authenticated and Progressive System Image Renovation

Sangho Lee

2023 USENIX Annual Technical Conference

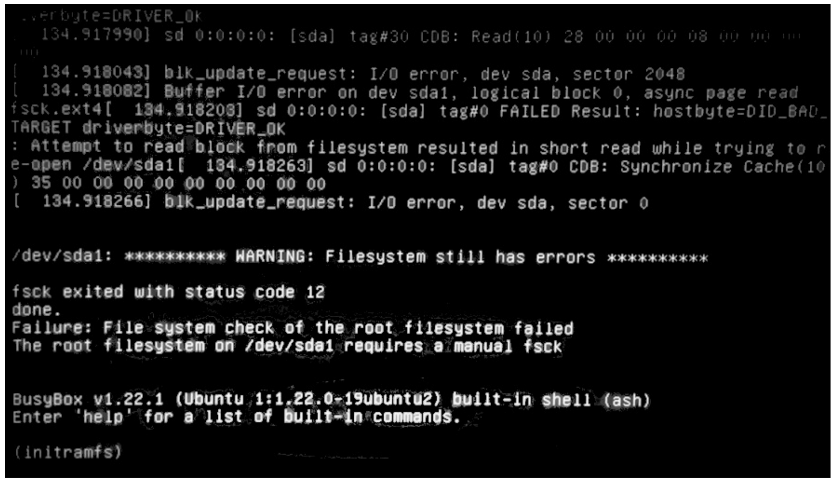
July 10, 2023

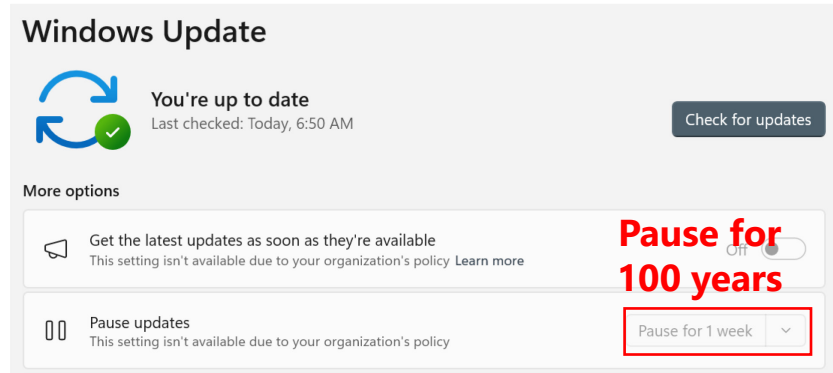
Devices or systems might be corrupt



* https://www.wirelesshack.org/fixes-for-your-device-is-corrupted-and-cannot-be-trusted.html
* https://answers.microsoft.com/en-us/windows/forum/all/recovery-your-pcdevice-needs-to-be-repaired/6d8a33b0-ccdb-43f7-b50c-1937aec79033

Why and how are systems corrupt?





Attacks/Malware

Software/hardware errors

Postponed updates

* https://securelist.com/wannacry-ransomware-used-in-widespread-attacks-all-over-the-world/78351/
* https://askubuntu.com/questions/972978/fsck-reports-that-filesystem-still-has-errors/

System recovery flow

A system (somehow) was **corrupt**.

System corruption is **detected**.

The system gets (forcefully*) **reset**.

A **recovery environment** repairs the system.

The system runs **normally**.

* M. Xu et al., "Dominance as a New Trusted Computing Primitive for the Internet of Things," IEEE S&P 2019.

What does a recovery environment do?

- Prepare a separate environment with recovery tools
- Completely recover corrupt system storage using a reference image
  - Download one from a remote location
  - Use one stored in a safe local location
- Verify the recovered system and restart it

Observation:
Recovery time ≈ system downtime

- Recovery environment is designed to be minimal.
  - Contain recovery-related tools only
  - Lack everyday programs, libraries, …
- System is effectively **down** during recovery.
- Delaying recovery lengthens system downtime.
  - Download a large reference image file (and decompress it)
  - Reimage the entire storage and verify it

Questionable ways to speed up recovery

- Use a stored system image (e.g., A/B partition)
  - Stored image can also be corrupted or outdated.
- Selectively fix corrupted files/blocks (delta recovery)
  - Difference calculation (e.g., rdiff) and scattered disk updates are slow.

Key idea: Progressive recovery

Defer the recovery of data blocks until they are needed

- A complete system image is not needed to start it.
  - Boot: A small part of it (i.e., kernel, initramfs) is required.
  - Execution: Some parts related to active tasks are required.

➢ **Selectively/partially** recover the system and make it **progressively** recover the remainder on demand

Key idea: Authenticated recovery

Verify every storage access to detect/recover invalid blocks

- Apps must be able to get valid data when they access system storage even if we defer its recovery.
  - Accessing invalid data blocks is prohibited.

➢Interpose storage access to always return valid data

- Verify a requested block with authenticated metadata
- Serve the block after recovery if it was corrupt or outdated

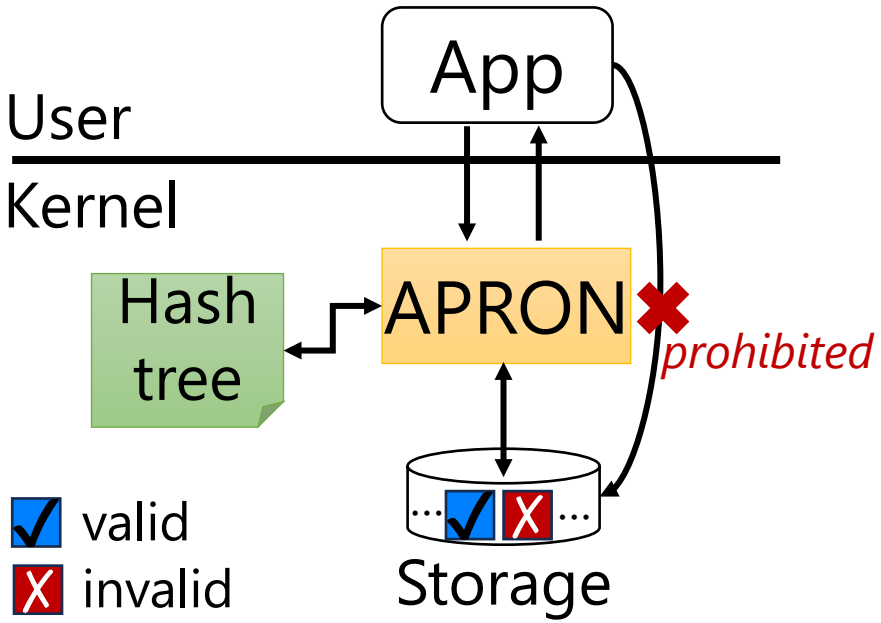
Image-based (immutable) operating system

- Management server builds an up-to-date system image and provisions it to managed devices.
- Operating system kernel enforces the read-only property of the system image.
- Store and manage mutable configuration and data in a separate place

Preparation and device initialization

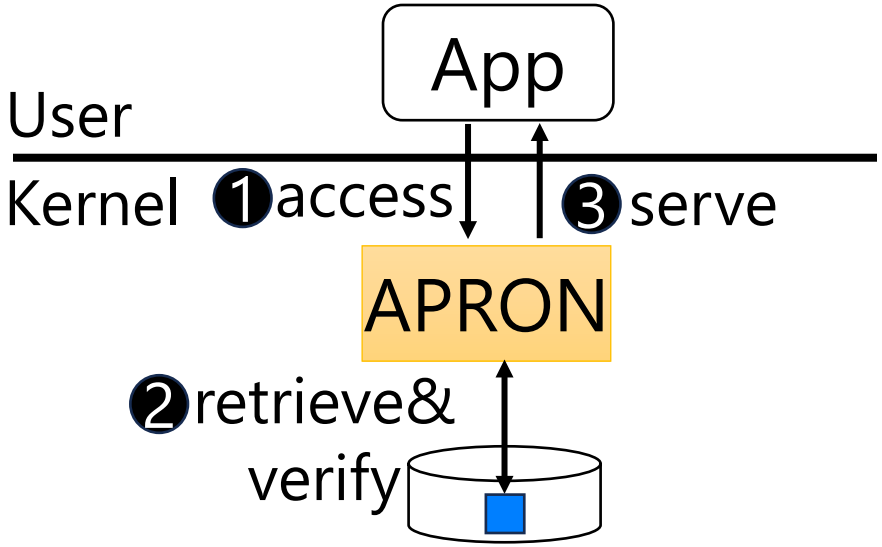
- Server
  - Calculate a hash tree over the latest system image
  - Sign its root hash concatenated with a version number
  - Serve the system image and signed metadata
- Device (initramfs)
  - Download and verify the latest metadata
  - Create a storage layer over system storage with the metadata

APRON storage layer

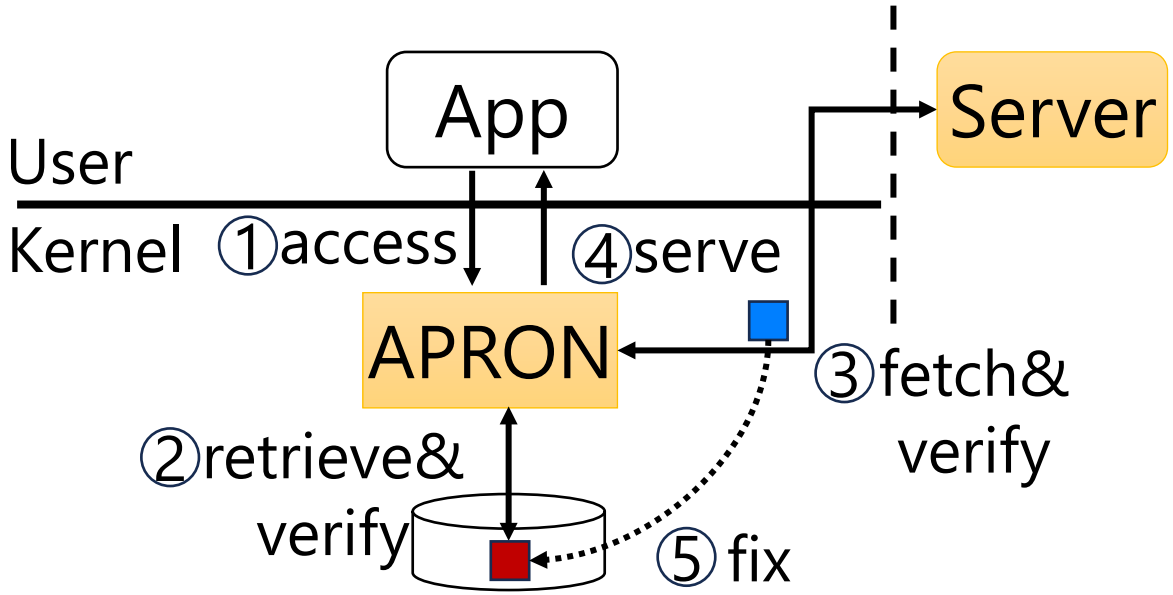


- Intervene with every access to system storage
- Verify each requested block using the hash tree
- Repair requested blocks if they are invalid (hash mismatch)

On-demand renovation (valid block)

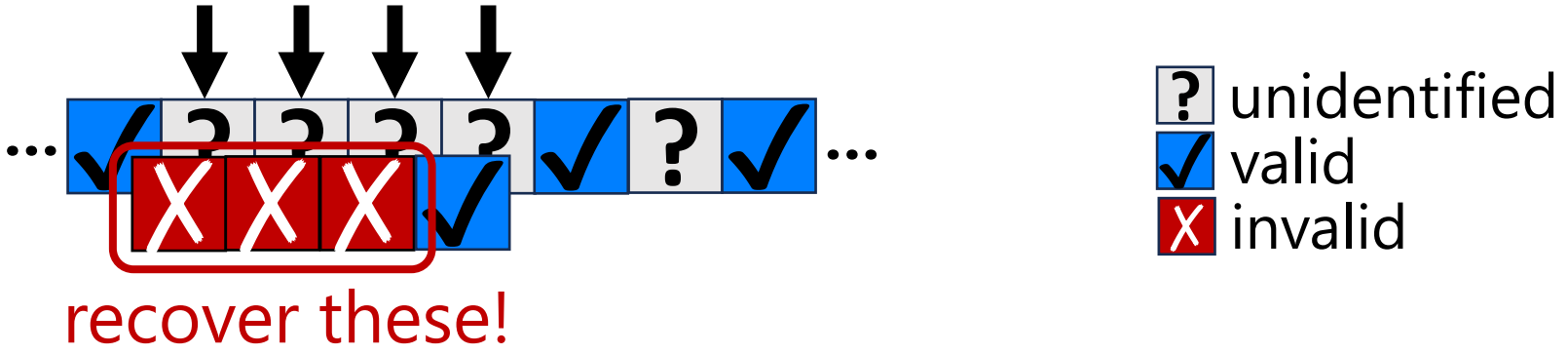


On-demand renovation (invalid block)



Background prefetcher

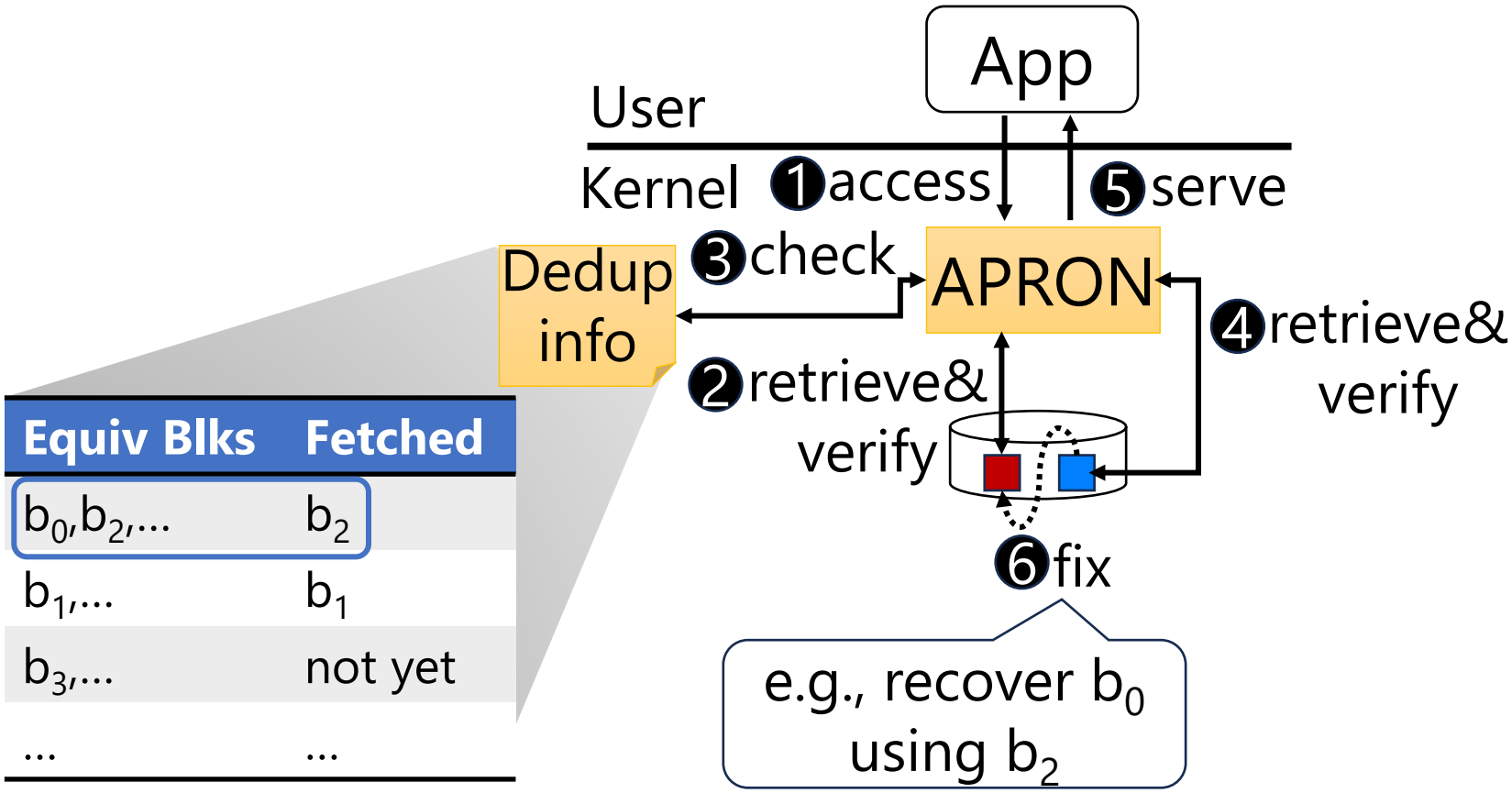
- Renovate non-yet requested blocks in background
  - Eventually recover the entire system storage
  - Wake up if there is no in-flight storage access
- Find and batch repair consecutive invalid blocks
  - Inspect unidentified blocks until it encounters a valid block
  - Recover the found invalid blocks together (for throughput)



In-place renovation with deduplication

- Fix requested blocks with equivalent blocks in storage
- Rely on static and dynamic deduplication information
  - Server pre-computes sets of equivalent blocks.
  - Device tracks whether it has fetched any block of each set.
- Fetch a remote block for recovery only if
  - It is unique; or
  - None of its equivalent block has been fetched.

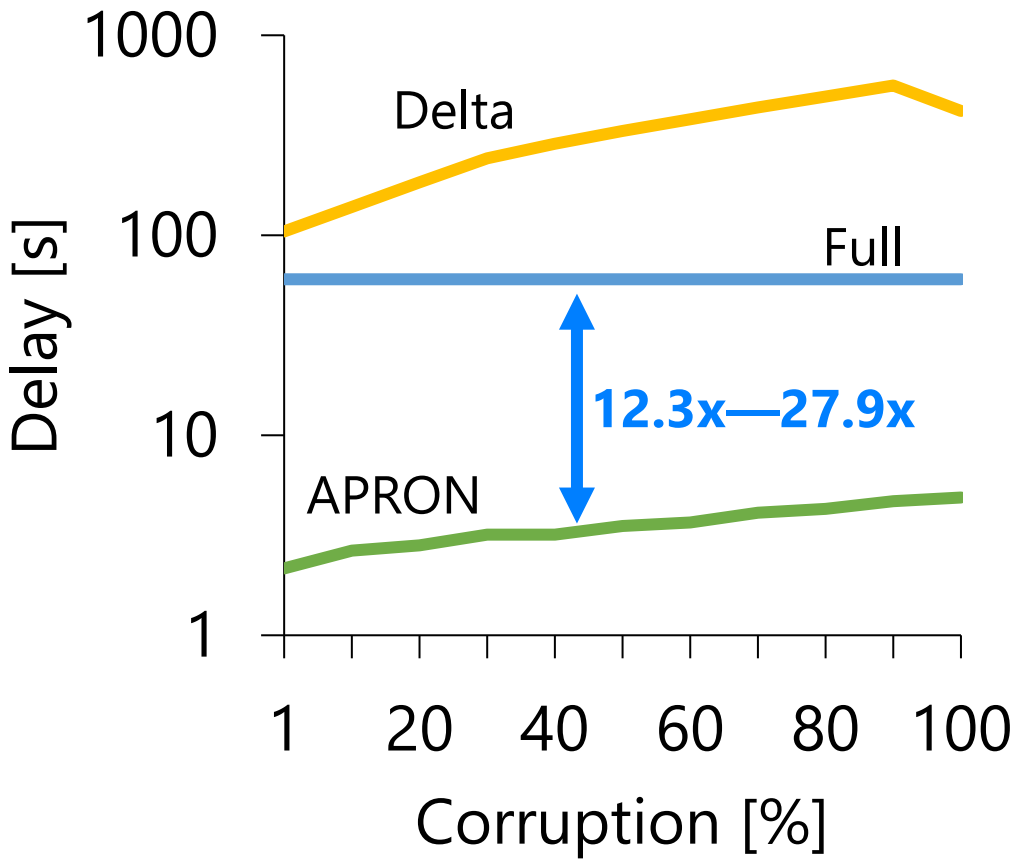
In-place renovation (invalid block)



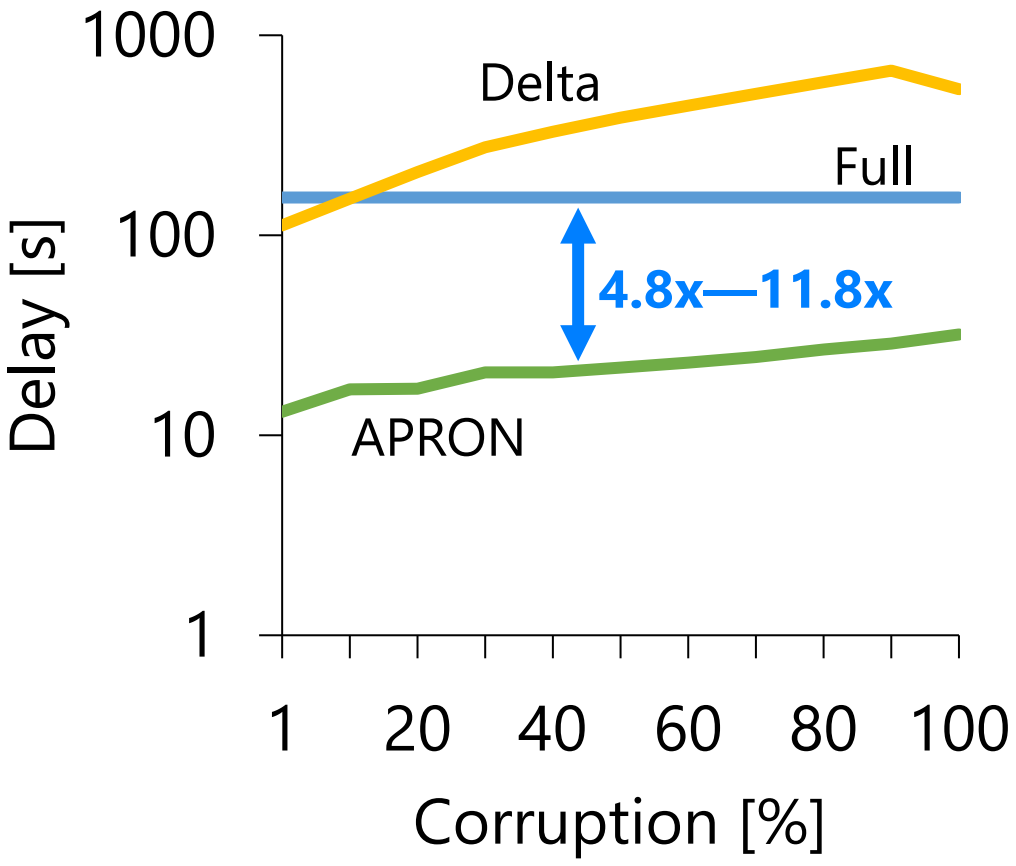
Evaluation setup

- Machineries
  - Client: Desktop CPU (six cores) & PCIe NVMe SSD
  - Fast-network server (1 Gbps): Desktop CPU (four cores) & SATA SSD
  - Slow-network server (100 Mbps): Server VCPU (two cores) & virtual SSD
- System image
  - 10 GiB of Ubuntu Server 20.04 installation
  - Randomly corrupt 1%—100% of it

System downtime

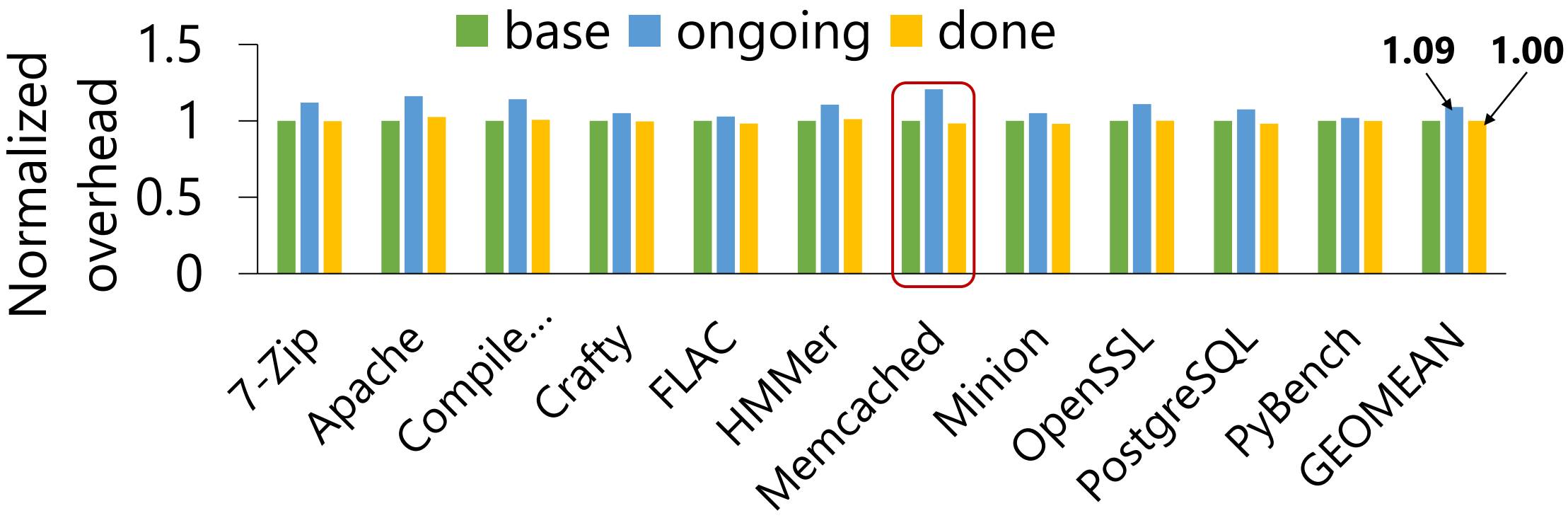


Fast server (1 Gbps)



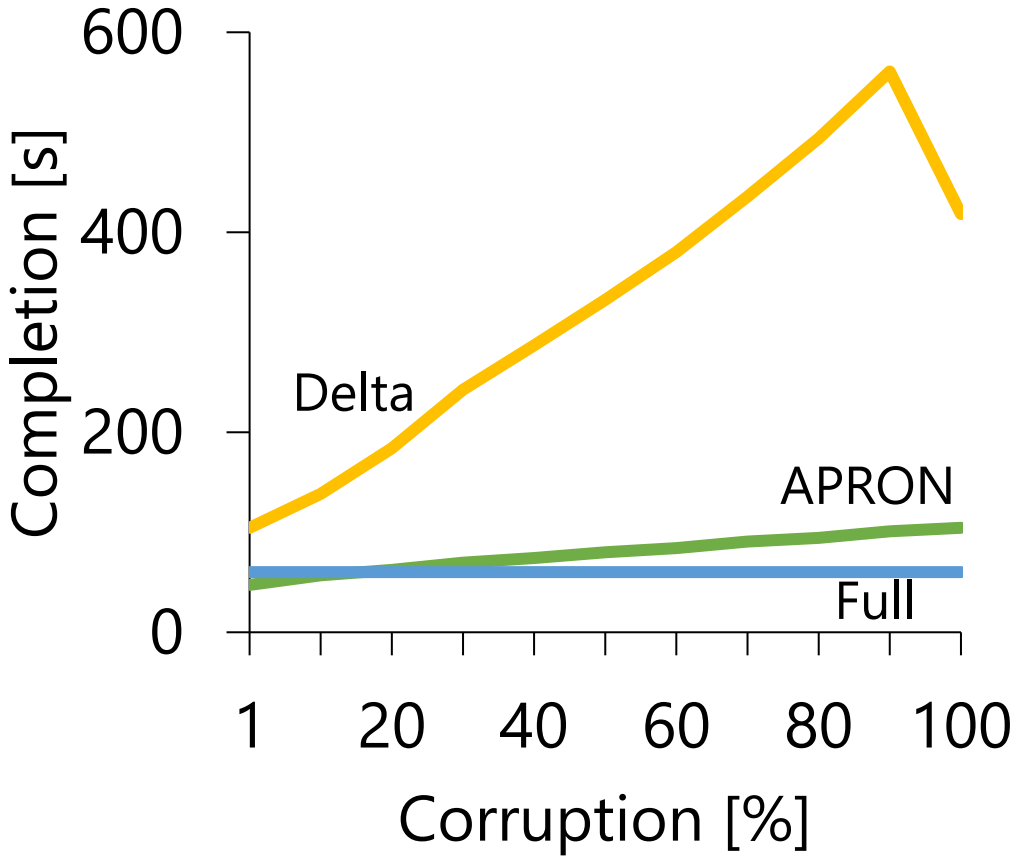
Slow server (100 Mbps)

Runtime overhead: Phoronix test suite

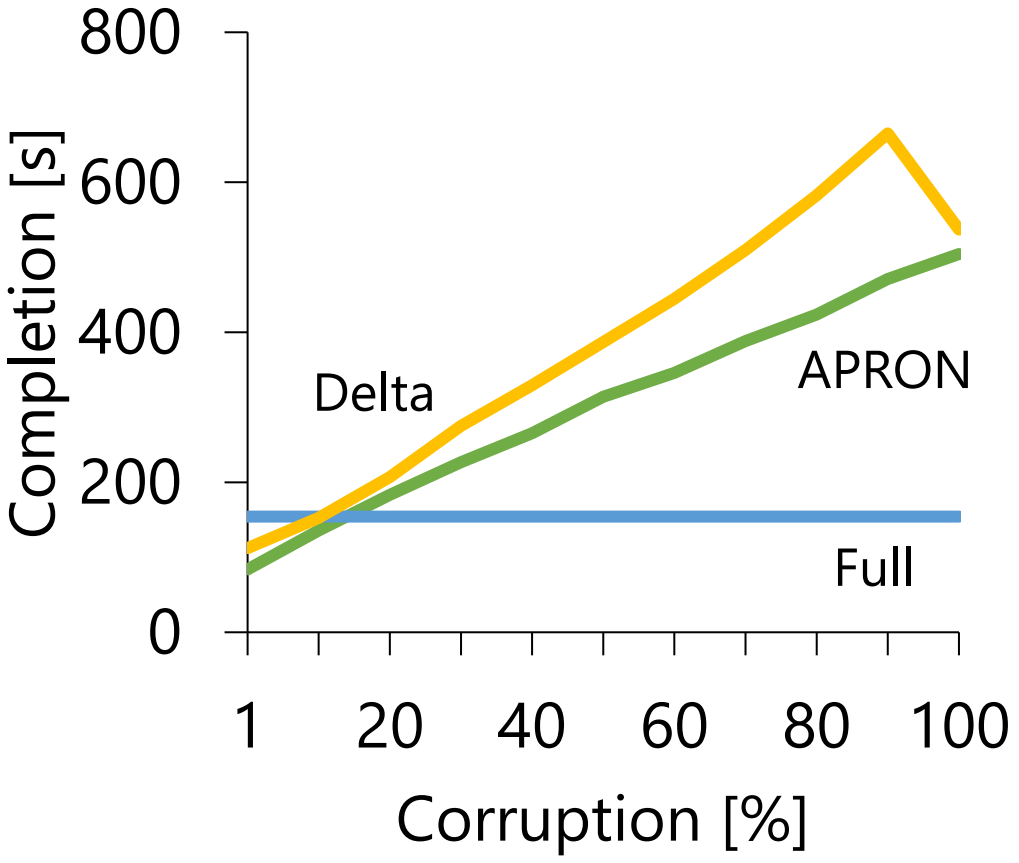


- Fully corrupted system storage and slow-network server
- Affect I/O-intensive workloads (e.g., Memcached: 21%)

Complete renovation time



Fast server (1 Gbps)



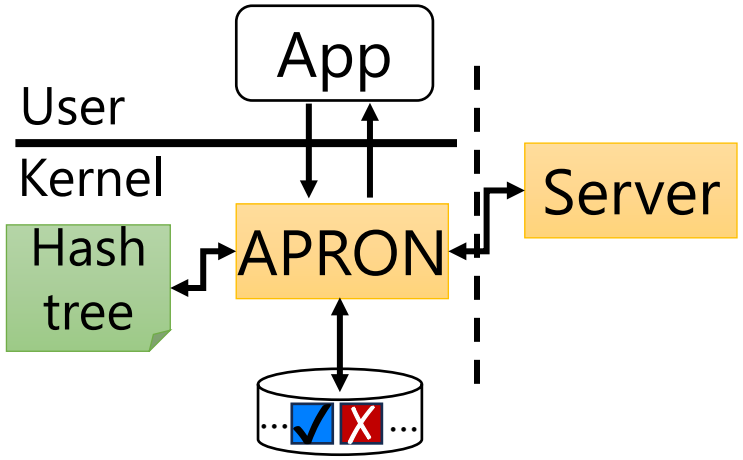
Slow server (100 Mbps)

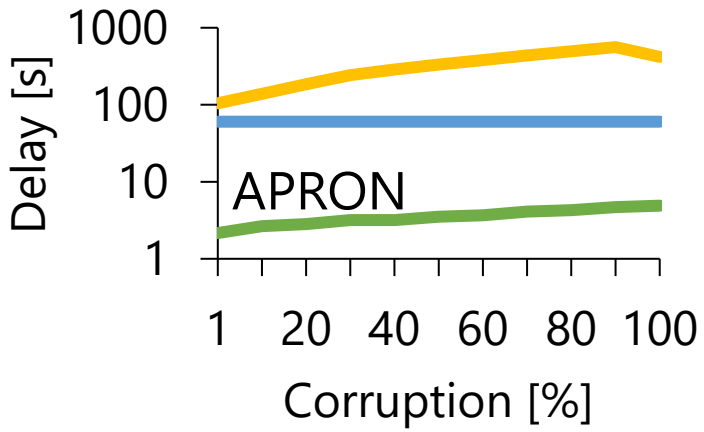
Conclusion

Slow recovery makes the system unavailable.

Securely defer recovery for instant system availability

Ensure short downtime and low runtime overhead





Sangho.Lee@microsoft.com

https://github.com/microsoft/APRON (soon)