ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ • ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING • UNIVERSITY OF IOANNINA

ΠΑΡΟΥΣΙΑΣΗ ΔΙΔΑΚΤΟΡΙΚΗΣ ΔΙΑΤΡΙΒΗΣ

ΗΜΕΡΟΜΗΝΙΑ: Παρασκευή, 25 Σεπτεμβρίου 2015

ΩΡΑ: 17:00

ΑΙΘΟΥΣΑ: Αίθουσα Σεμιναρίων
Κτήριο Τμήματος Μηχανικών Η/Υ & Πληροφορικής

ΟΜΙΛΗΤΗΣ: **Ανδρομάχη Χατζηελευθερίου**

Θέμα

«Fast and Efficient Durable Storage for Local And Distributed Filesystems»

Επταμελής Εξεταστική Επιτροπή:

1. **Στέργιος Αναστασιάδης**, Αναπληρωτής Καθηγητής του Τμήματος Μηχανικών Η/Υ & Πληροφορικής, Πανεπιστήμιο Ιωαννίνων (*επιβλέπων*).
2. **Ευαγγελία Πιτουρά,** Καθηγήτρια του Τμήματος Μηχανικών Η/Υ & Πληροφορικής, Πανεπιστήμιο Ιωαννίνων.
3. **Γεώργιος Μανής,** Επίκουρος Καθηγητής του Τμήματος Μηχανικών Η/Υ & Πληροφορικής, Πανεπιστήμιο Ιωαννίνων.
4. **Κωνσταντίνος Μαγκούτης,** Επίκουρος Καθηγητής του Τμήματος Μηχανικών Η/Υ & Πληροφορικής, Πανεπιστήμιο Ιωαννίνων.
5. **Αλέξης Δελής,** Καθηγητής του Τμήματος Πληροφορικής & Τηλεπικοινωνιών, Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών.
6. **Toni Cortes,** Αναπληρωτής Καθηγητής του Department of Computer Architecture, Universitat Politecnica de Catalunya (UPC), Spain.
7. **Peter J. Varman,** Καθηγητής του Department of Electrical & Computer Engineering, Rice University, U.S.A.

Περίληψη

The increasingly large amount of data that needs to be stored and processed in modern cloud applications requires highly scalable storage systems. In a typical cloud environment, the storage stack consists of multiple tiers. Frontend machines are mainly responsible for temporarily caching data, and backend machines provide persistent storage. Additionally, data is redundantly stored among multiple servers for high availability in case of failures. However, the co-location of multiple workloads in a shared physical infrastructure introduces several design challenges related to the performance, resource efficiency and data durability. In this thesis, we argue that the inherent characteristics of multi-tier virtualized environments necessitate the fresh reconsideration of the I/O path. Across different tiers of the storage stack, we investigate the tradeoff between consistency and resource efficiency, aiming to provide improved durability and high performance at relatively low resource overhead.

Initially, we focus on the storage backend aiming to combine improved local filesystem consistency with high performance and efficient storage bandwidth utilization. Synchronous small writes are commonly used to safely log recent state modifications for fast crash recovery. Demanding systems usually dedicate separate devices to logging for adequate performance during normal operation and redundancy during state reconstruction. Nevertheless, storage stacks enforce page-sized granularity in data transfers from memory to disk. As a result, they consume excessive storage bandwidth to handle small writes, which hurts performance. The problem worsens, as filesystems often handle multiple concurrent streams, which effectively generate random I/O traffic. In a local filesystem, we rely on journaling of both data and metadata blocks in order to achieve their safe transfer to disk at sequential disk throughput and low latency. We propose the design of two new mount modes, wasteless journaling and selective journaling. Wasteless journaling coalesces multiple concurrent subpage writes into page-sized journal blocks. Instead, selective journaling selectively journals data updates below a write threshold, and transfers the rest directly to the filesystem. We implement a functional prototype of our design over a widely-used local filesystem. Across a wide range of microbenchmarks and application-level workloads over standalone servers and a multi-tier networked system, we demonstrate that the proposed modes preserve filesystem consistency, and provide improved operation throughput along with reduced write latency and recovery time, at low storage bandwidth overhead.

We further examine the implications among performance, resource efficiency, and durability in scalable storage systems by focusing on the client-side frontend of the storage stack. Hardware consolidation in the datacenter occasionally leads to scalability bottlenecks due to the heavy utilization of critical resources, such as the shared network bandwidth. Host-side caching on durable media is already applied at the block level in order to reduce the load of the storage backend. However, block-level caching is often criticized for added overhead and restricted data sharing across different hosts. During client crashes, writeback caching can also lead to unrecoverable loss of written data that was previously acknowledged as stable. We improve the durability of shared storage in the datacenter by supporting journaling at the kernel-level client of a well-known object-based distributed filesystem. Storage virtualization at the file interface allows us to achieve clear consistency semantics across data and metadata blocks, support native file sharing between clients over the same or different hosts, and provide flexible configuration of the time period during which the data is durably staged at the host side. Over a prototype implementation, we demonstrate improved operation throughput at reduced disk and network bandwidth utilization for specific durability, across multiple microbenchmarks, application-level workloads, and real-world applications on top of a local cluster setup and a public cloud environment.