

POSIX is Dead! Long Live... errr... What Exactly?

<u>Erez Zadok,</u>	Dean Hildebrand,	Geoff Kuenning, and	Keith Smith
Stony Brook University	IBM Research - Almaden	Harvey Mudd College	NetApp



July 10, 2017

POSUX talk (HotStorage'17 WACI)

1



The Problem with POSIX (1)

- POSIX is ~30 years old
 - ◆ Huge advancements in computing
 - ◆ Deeper, more complex storage stack
- Inefficient use of resources
 - ◆ Designed for serial/sync data access
 - ◆ Today: faster, more complex, heterogeneous hardware
 - Parallelism, clouds

July 10, 2017

POSUX talk (HotStorage'17 WACI)

2



The Problem with POSIX (2)

- Security: Huh? What's that?
 - ◆ E.g., TOCTTOU bugs
- Assumes files and namespaces
 - ◆ Hard to manage many files
- Byte-level access
 - ◆ Overkill for Objects?



July 10, 2017

POSUX talk (HotStorage'17 WACI)

3



How Do You Solve a Problem Like...

- Abolish POSIX!
- New API
 - ◆ Works with files, objects, extents, etc.
- Minimal set of native calls
 - ◆ E.g., really need symlinks/hardlinks?!
- Security from day one
- Efficiency
 - ◆ Asynchrony, compounding, API v2



July 10, 2017

POSUX talk (HotStorage'17 WACI)

4



Increasing Efficiency (1)

- Compounding (bulk updates)
 - ◆ Pack multiple request in one message
 - ◆ Saves latency, fewer context switches
 - ◆ Multiple failure semantics
 - First-fail
 - Continue-on-fail
 - Transactional
 - ◆ Simple if-then-else and looping syntax

[FAST'17, HOTOS'03]

July 10, 2017

POSUX talk (HotStorage'17 WACI)

5



Increasing Efficiency (2)

- Asynchrony everywhere
 - ◆ Encourages more parallel "thinking"
 - ◆ CPU speeds vs. #cores
- API changes
 - ◆ Give a "handle" to operate on a collection of files/objects
 - Locking, leases/delegations, recalls
 - ◆ Avoid passing same info repeatedly
 - E.g., /bin/cp passes same pathname
 - ◆ Improves security and reliability

[FAST'17, HOTOS'03]

July 10, 2017

POSUX talk (HotStorage'17 WACI)

6



Implementation

- Low-level vs. high-level compounding API
 - ◆ Low-level: allow compounding anything
 - ◆ High-level: useful APIs to developers
- Offer begin/end APIs
 - ◆ Mark code segments for compounding
 - ◆ Needs compiler/language support, code-analysis

July 10, 2017

POSIX talk (HotStorage'17 WACI)

7



Stony Brook University



Transitioning

- Implement most common requests first
 - ◆ e.g., read/write-whole-file, copy-file, rename-multiple-objects, rm-*
- Offer transactional semantics early on
 - ◆ Requires support all the way through the storage/OS/network stack



July 10, 2017

POSIX talk (HotStorage'17 WACI)

8



Stony Brook University



POSIX is Dead! Long Live... errr... What Exactly?

Q&A

Erez Zadok,

Stony Brook University

Dean Hildebrand,

IBM Research - Almaden

Geoff Kuenning, and

Harvey Mudd College

Keith Smith

NetApp



July 10, 2017

POSIX talk (HotStorage'17 WACI)

9



Stony Brook University

