Rethink the Sync!

Edmund B. Nightingale
Kaushik Veeraraghavan
Peter M. Chen
Jason Flinn

University of Michigan

Sync or Async?

- Desktop file systems typically use asynchronous I/O
- Advantages of synchronous file I/O:
  - Cleaner abstraction
  - Any output seen by user is durable
  - Application programming is easier
- Disadvantage of synchronous file I/O: Slow!
- Need new model: visible synchrony

Visible Synchrony

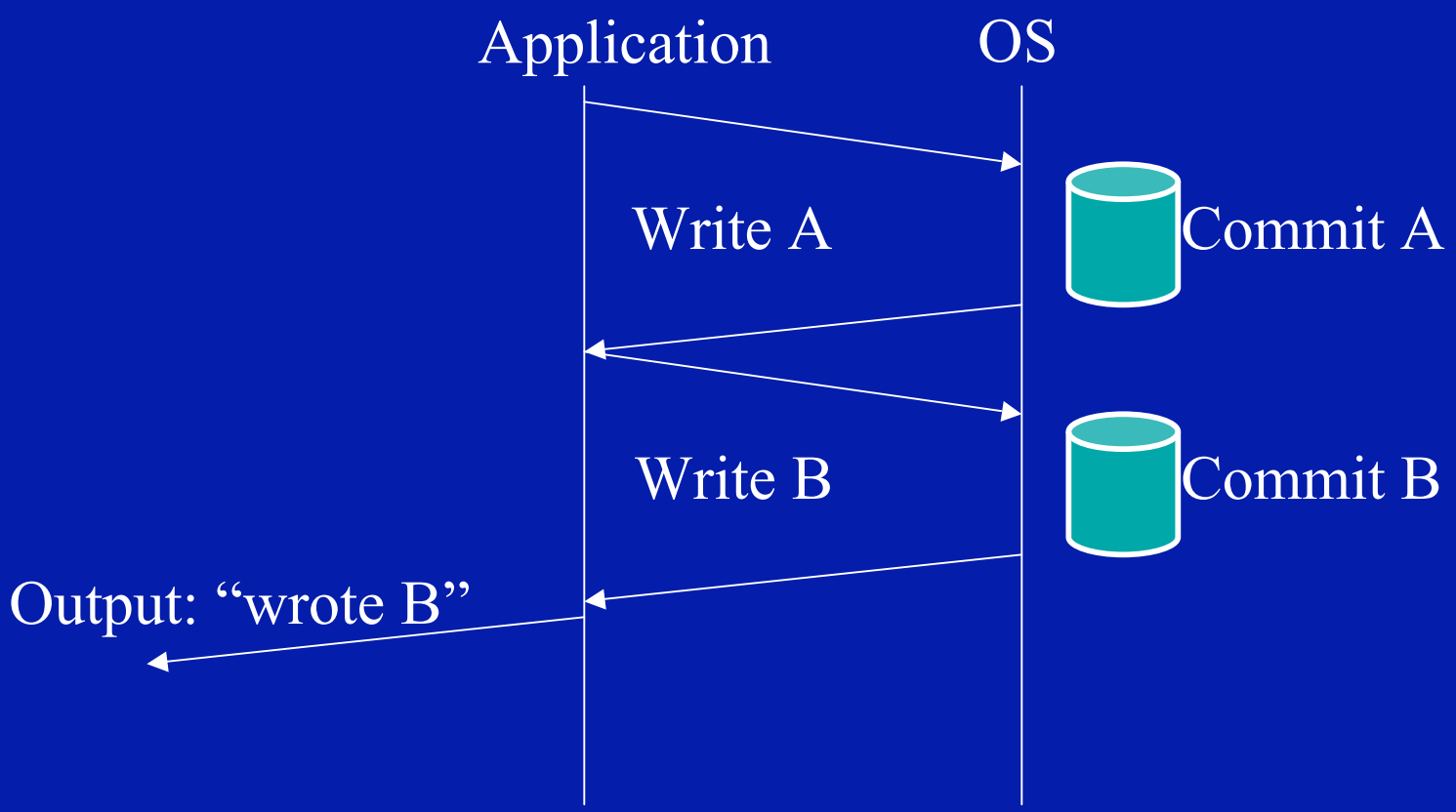
- Synchronous file operation guarantees:
  - Ordering: if A happens before B, effects of B not visible unless the effects of A also visible.
  - Durability: any operation that is observed to complete is already committed to disk.
- Current OSes provide guarantees to applications.
  - Do not return from system call until data committed.

Idea: provide guarantees to external observers instead!

Ordering and Durability

- Ordering: commit operations in temporal order
  - Use ext3 in data journaling mode.
- Durability:
  - File system operations return immediately but taint process.
  - OS buffers (uncommitted) output from tainted processes.
  - When data commits, OS removes taint, releases output.
  - OS tracks taint spread through IPC, shared memory, etc.

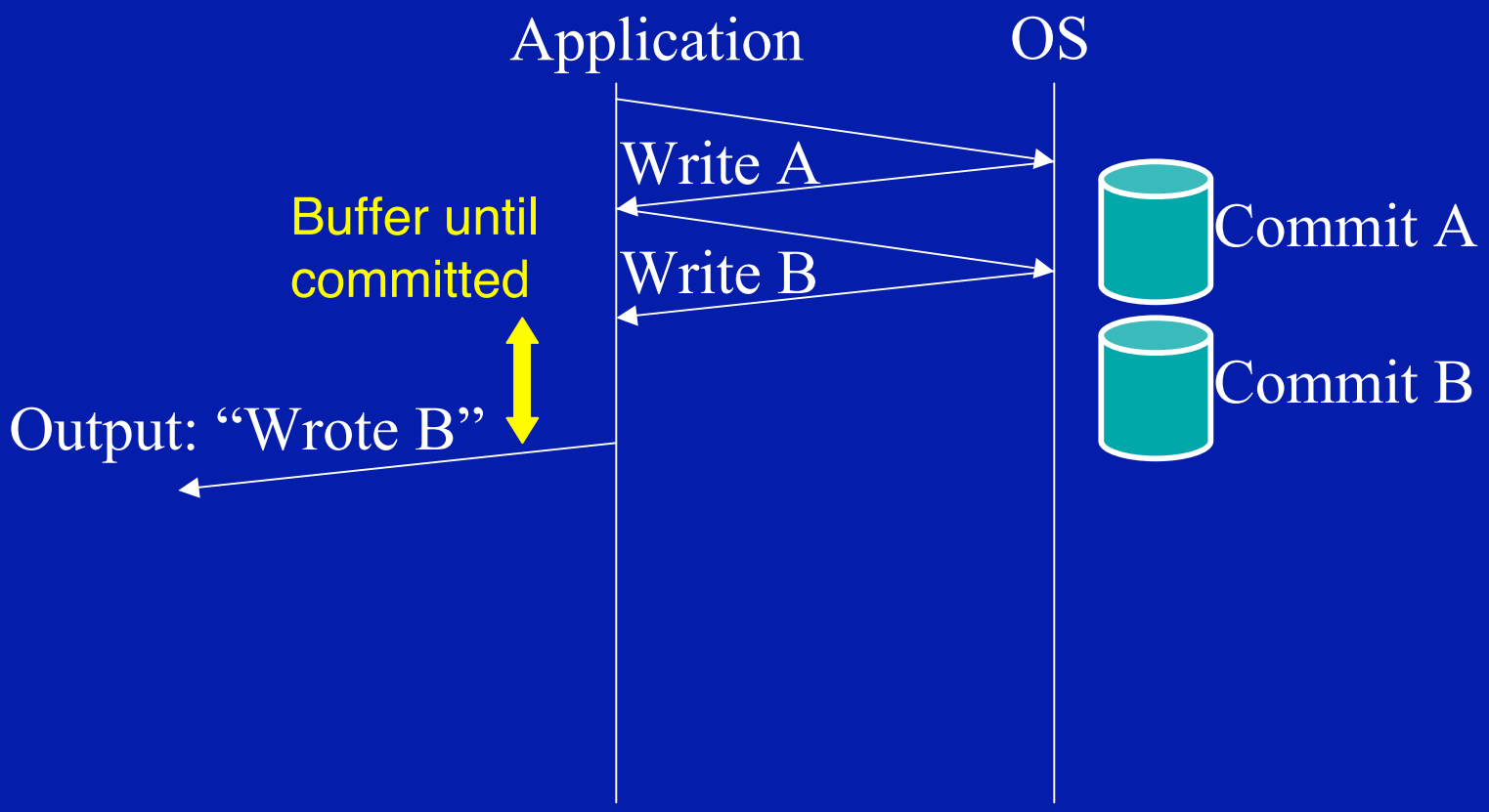
Synchronous I/O



Slow! Many synchronous disk writes

Rethink the Sync!

Visibly Synchronous I/O



Much faster (with a few more tricks)

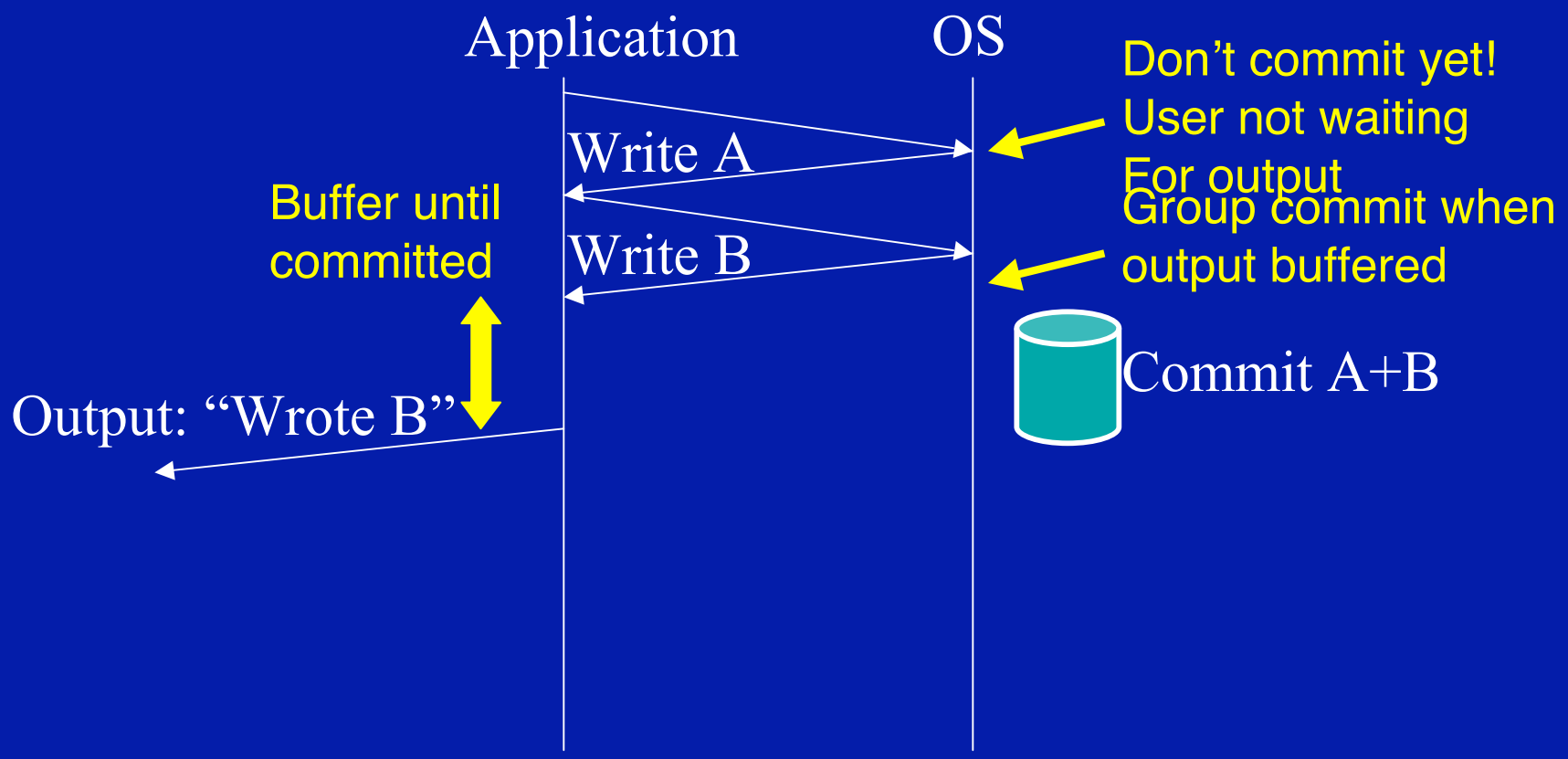
Rethink the Sync!

Results: Linux 2.4 kernel

	Async	Sync (default)	Sync (barriers)	Visible Synchrony
Durable?	No	Not power loss	Yes	Yes
after fsync?	No	Not power loss	Yes	Yes
Apache Build	1:46	2:20	21:27	1:50
(overhead)	---	(59%)	12x	(4%)
PostMark	8.1 s.	132 s.	1661 s.	8.8 s.
(overhead)	---	16x	204x	(8%)

Visible synchrony: durability with low overhead!
Users see behavior equivalent to synchronous I/O
No application modification required.

Latency vs. Throughput



Idea: Optimize for latency or throughput based upon visibility.