

The path to using AIO in Postgres

Andres Freund
PostgreSQL Developer & Committer

Email: andres@anarazel.de

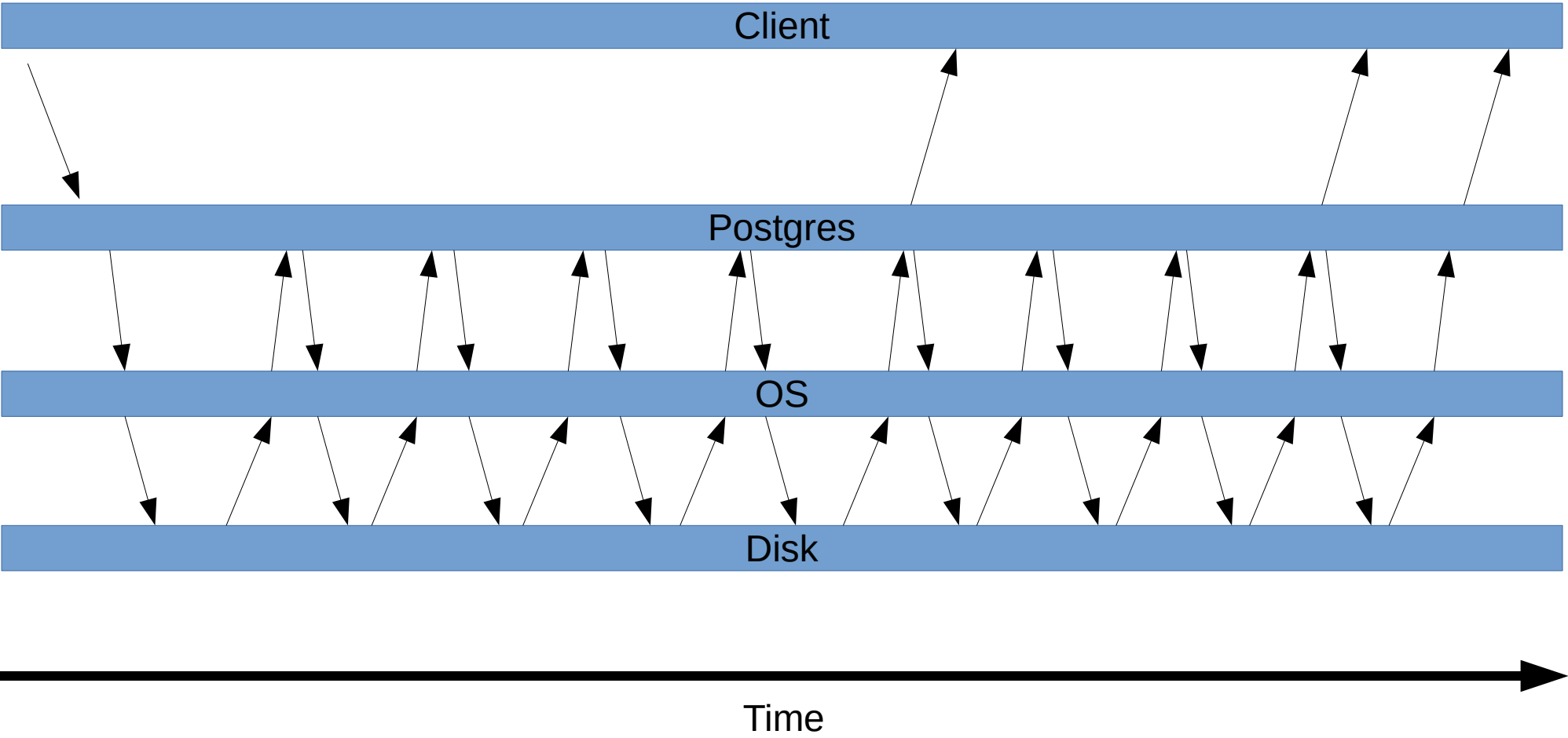
Email: andres.freund@microsoft.com

anarazel.de/talks/2023-12-14-pgconf-eu-path-to-aio/path-to-aio.pdf

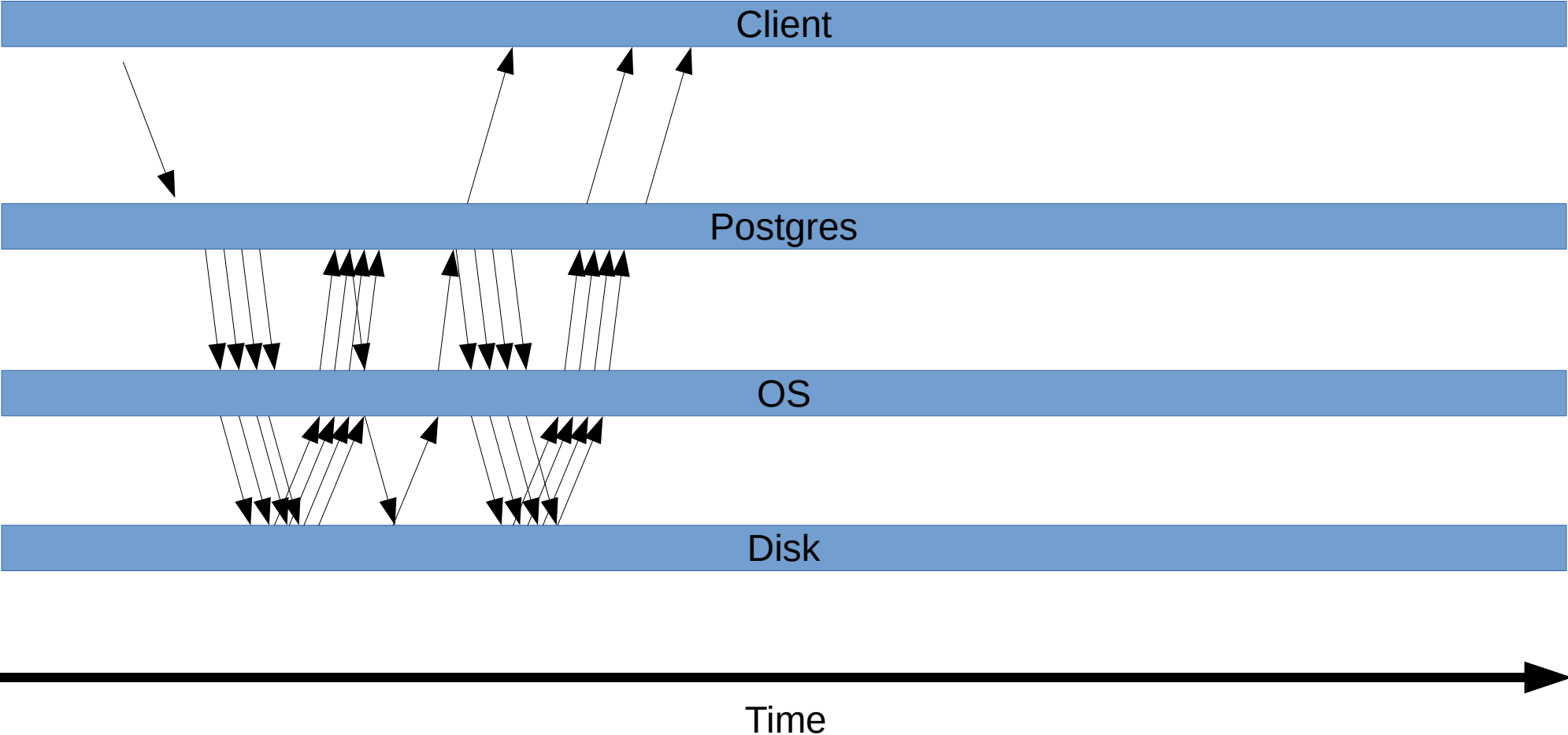
It's Long

It's Hard

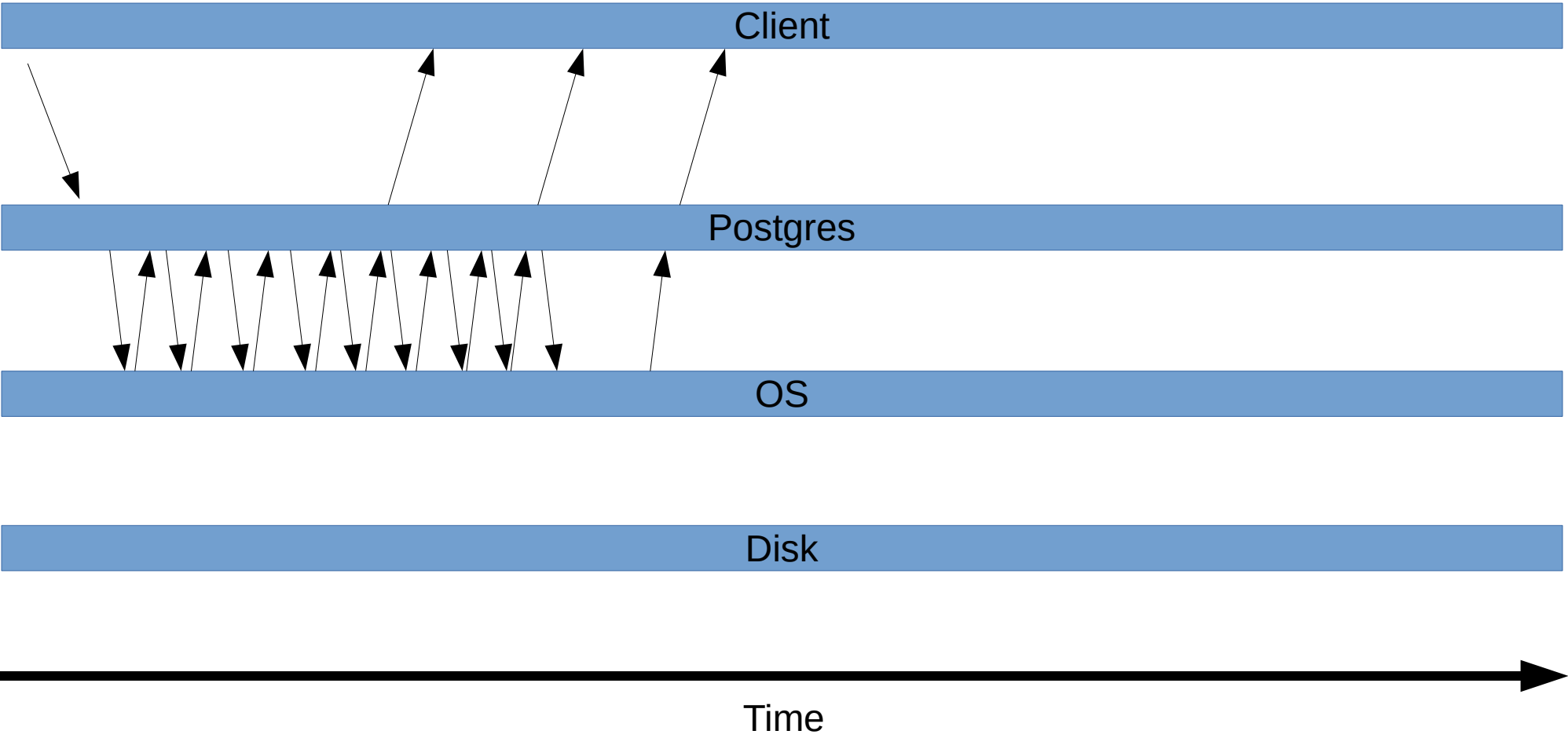
Reads: synchronous, not cached



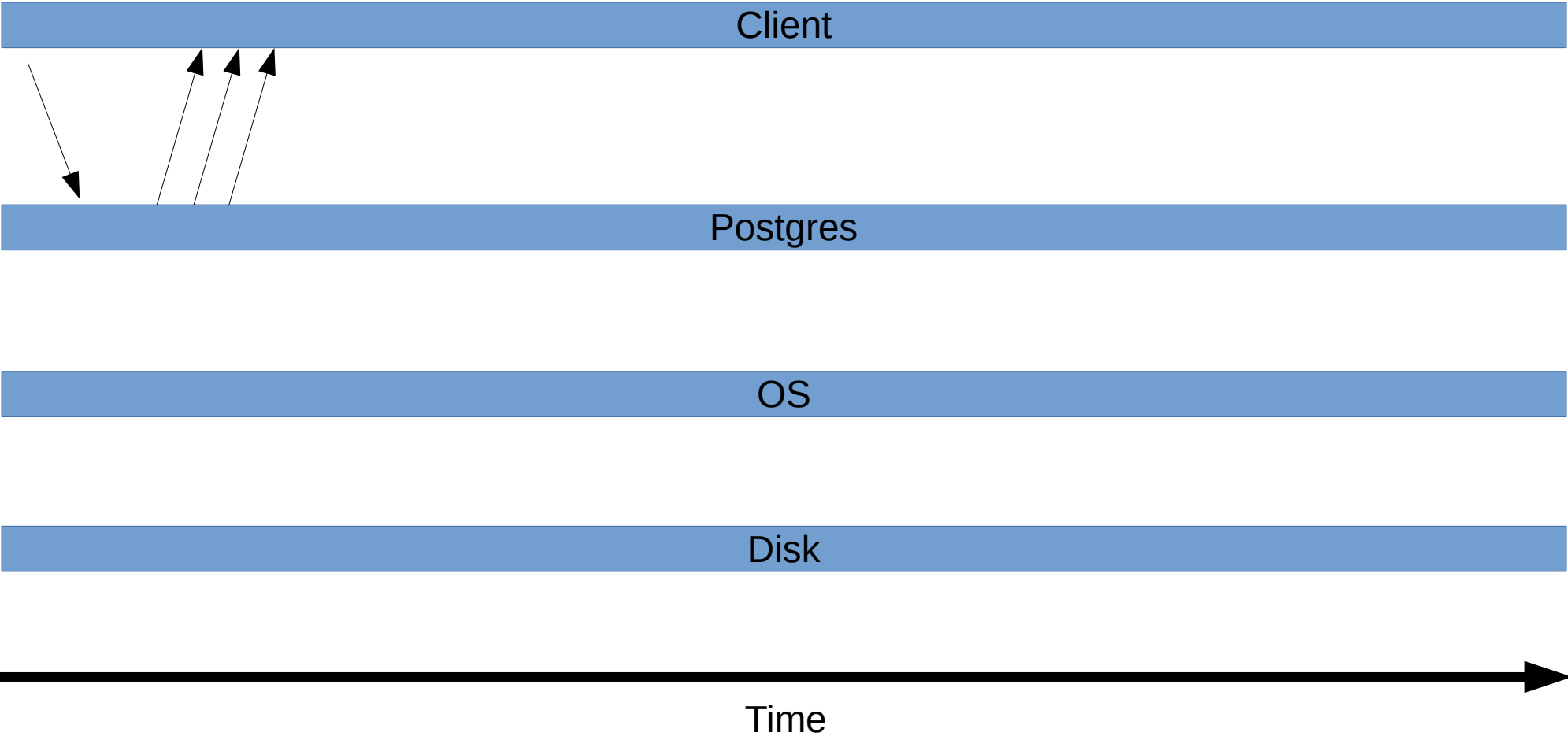
Reads: asynchronous, not cached



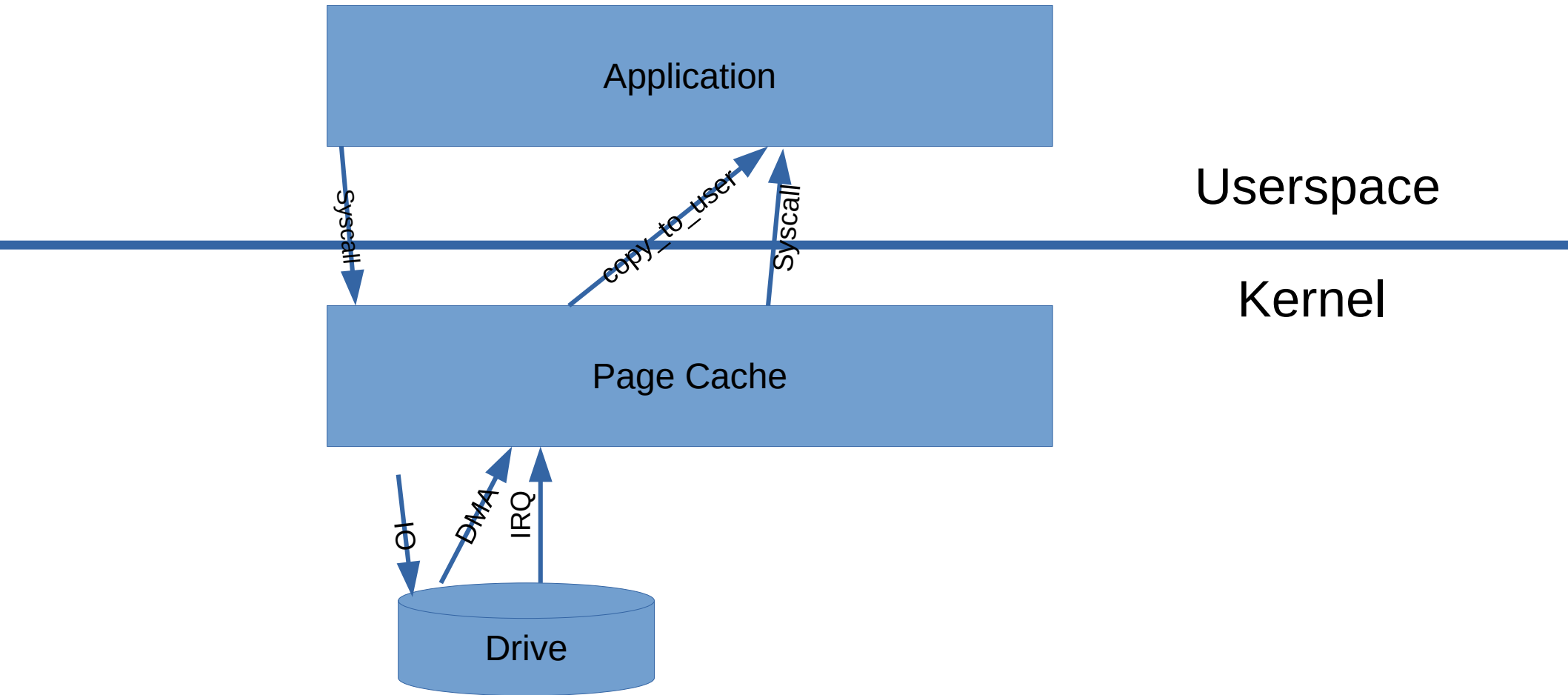
Reads: synchronous, OS cached



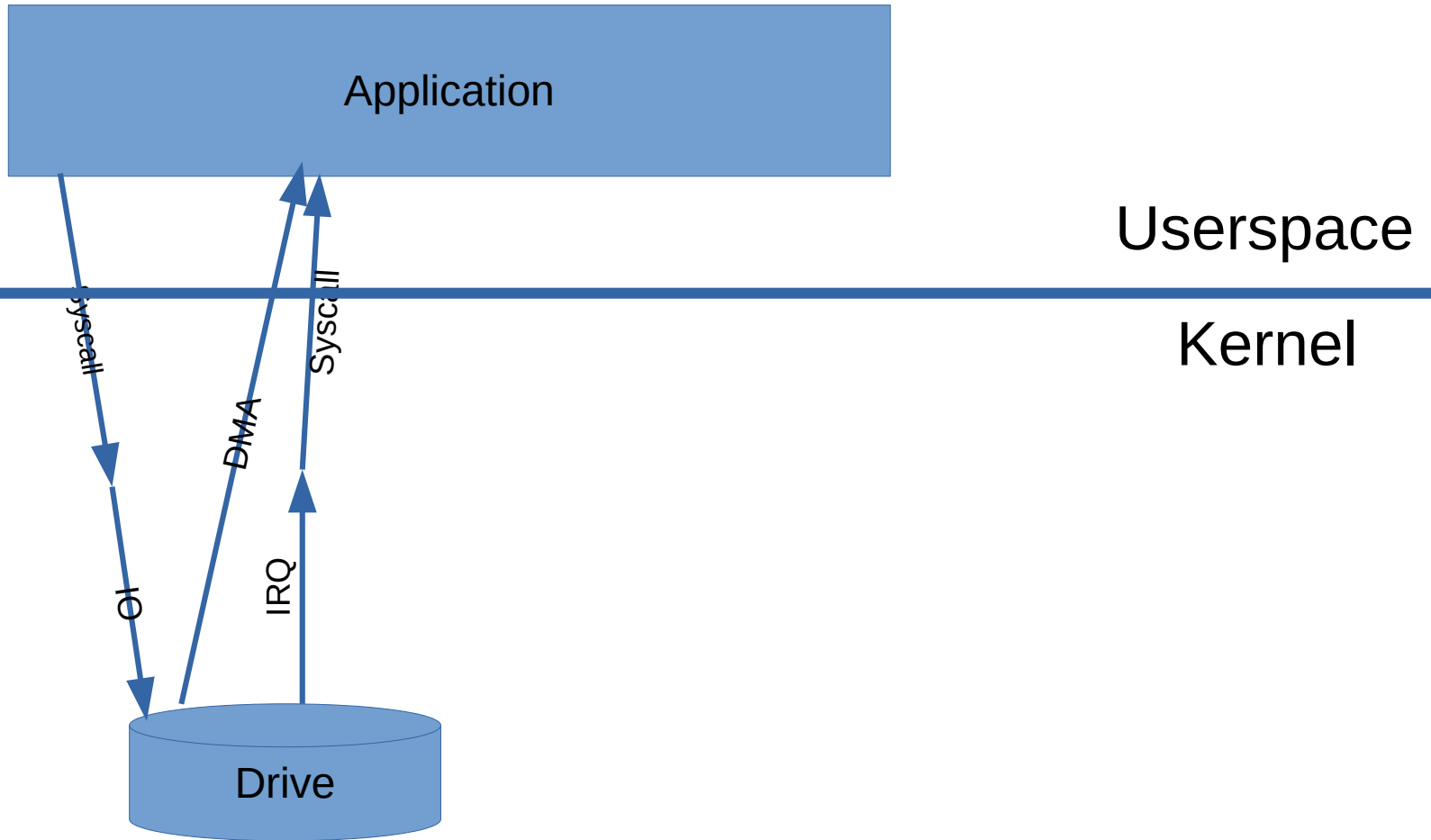
Reads: synchronous, postgres cached



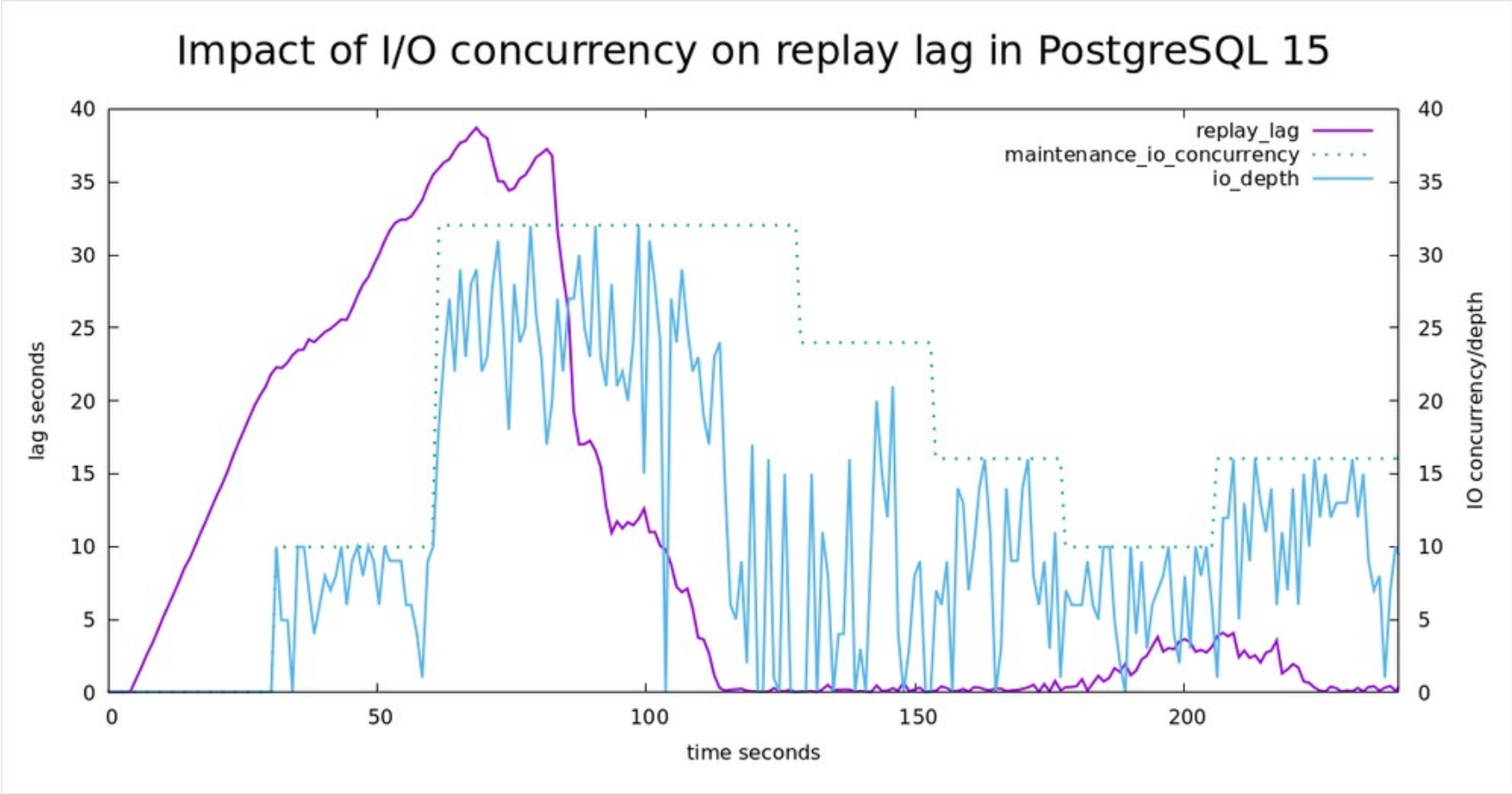
Buffered read()



Direct IO (DIO) read()



15: Recovery Prefetching (Thomas Munro)

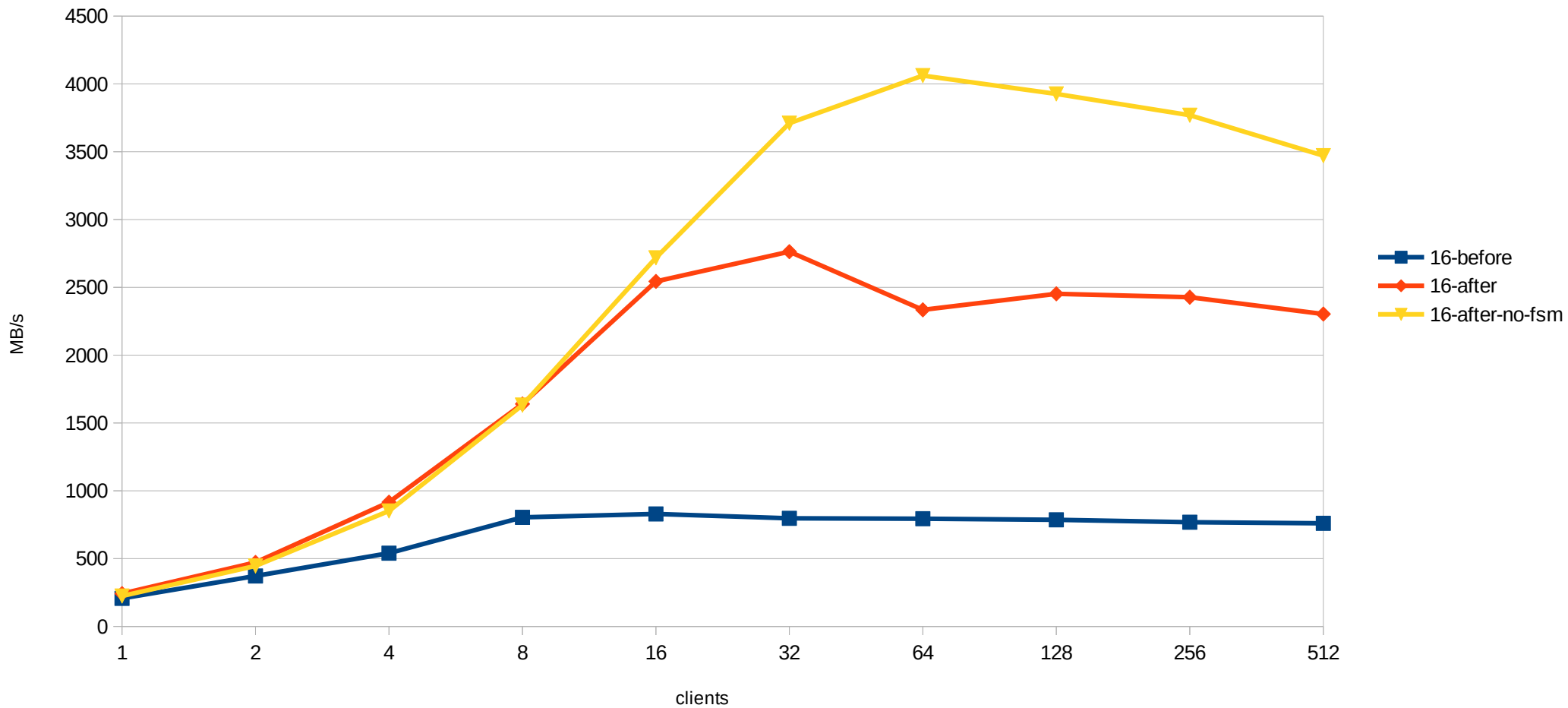


16: Bulk Relation Extension, Buffer Replacement

- Infrastructure for multiple “in progress”
Buffers
- Buffer Replacement – “Get a free buffer
when there are no free ones”
- Relation Extension - “Making a table bigger”

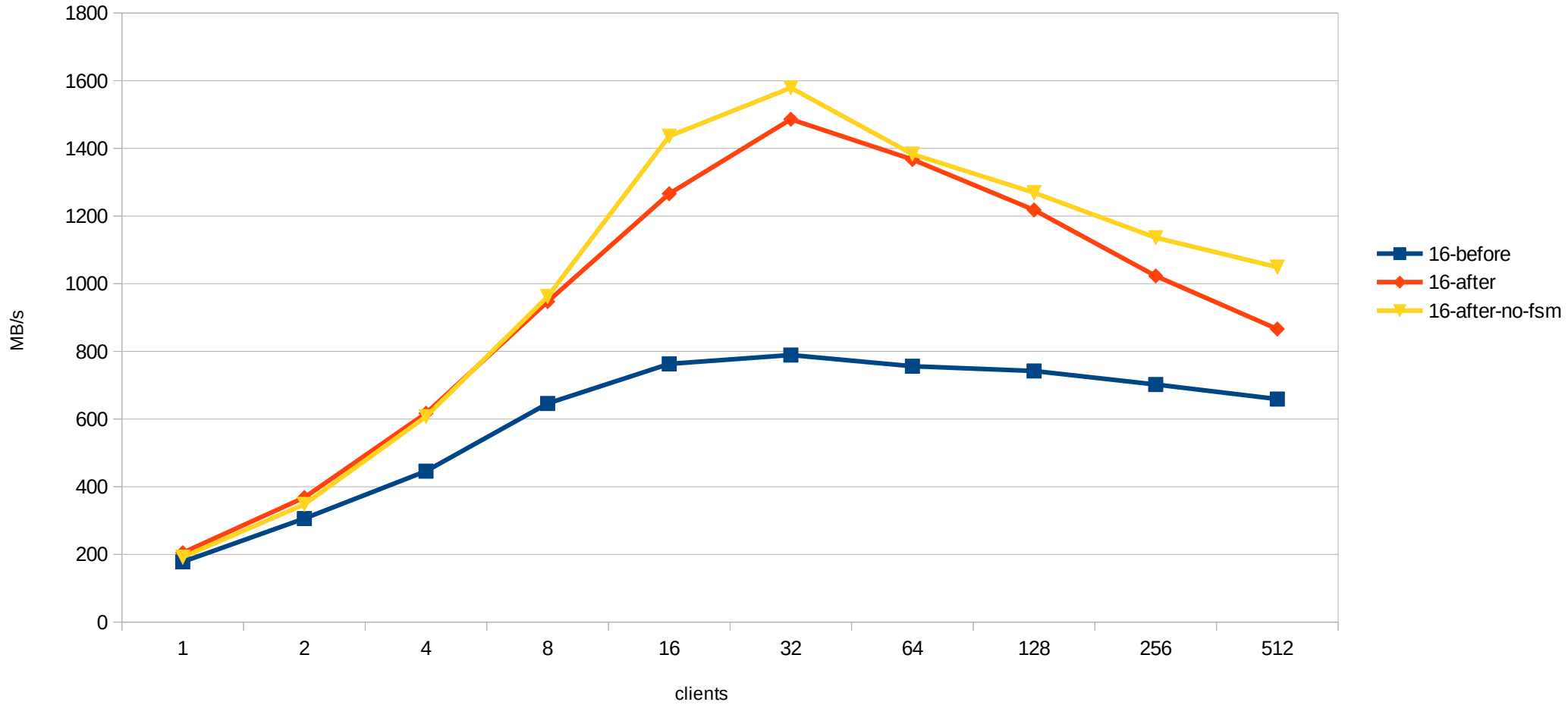
COPY into unlogged table

small files, ~10GB total, fits into s_b, 20c/40t machine



COPY into logged table

small files, ~10GB total, fits into s_b, 20c/40t machine



16: “Add debug_io_direct setting for developer usage”

- Can be set to data, wal, wal_init
- **NOT RECOMMENDED FOR PRODUCTION**
- “data”
 - relation IO
 - is disastrous for performance
- “wal”
 - can already show benefits, particularly with wal_sync_method=open_datasync
- “wal_init”
 - creation of new WAL files

17?: Streaming Read Abstraction

- Simple interface for most read IO
- Initially no AIO, just “fadvise” style prefetching
- Main Goal: Parallelize development
- Minor Goal: Small performance gains
- Convert some users to new interface
- See

17-18 ??: AIO infrastructure

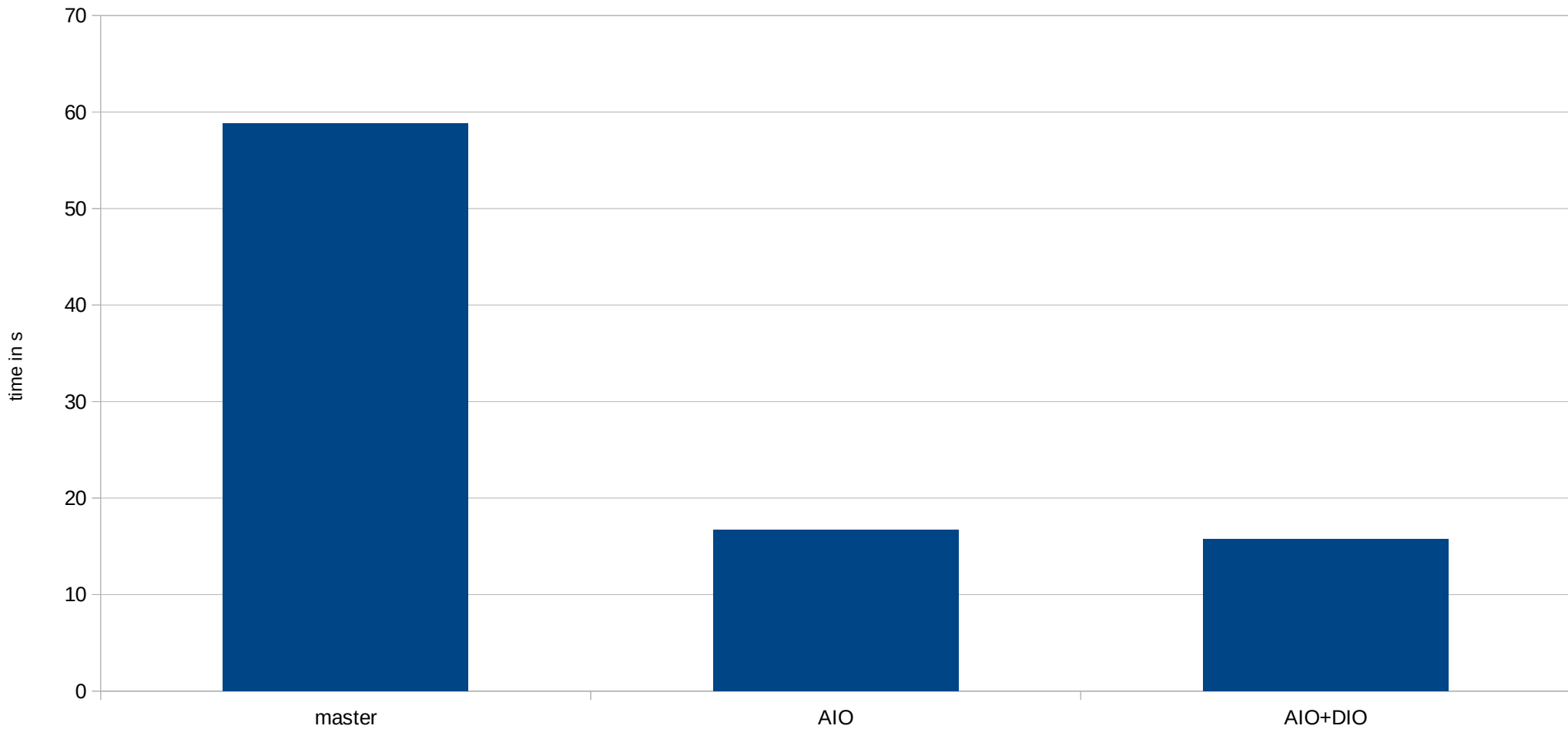
- `io_method=(worker|io_uring|posix_aio)`
- goal: can use AIO infrastructure without loss of performance, even when no AIO support present, to avoid duplicating code
- Not yet used

17-18 ?: sequential scans

- Problem:
 - Only use OS readahead
 - Double Buffering
 - Can get confused (skipped blocks, segments)
 - Not guaranteed to be present on all OSs
 - OS readahead doesn't know workload / not aggressive enough
 - Simple patch due to “streaming read” interface

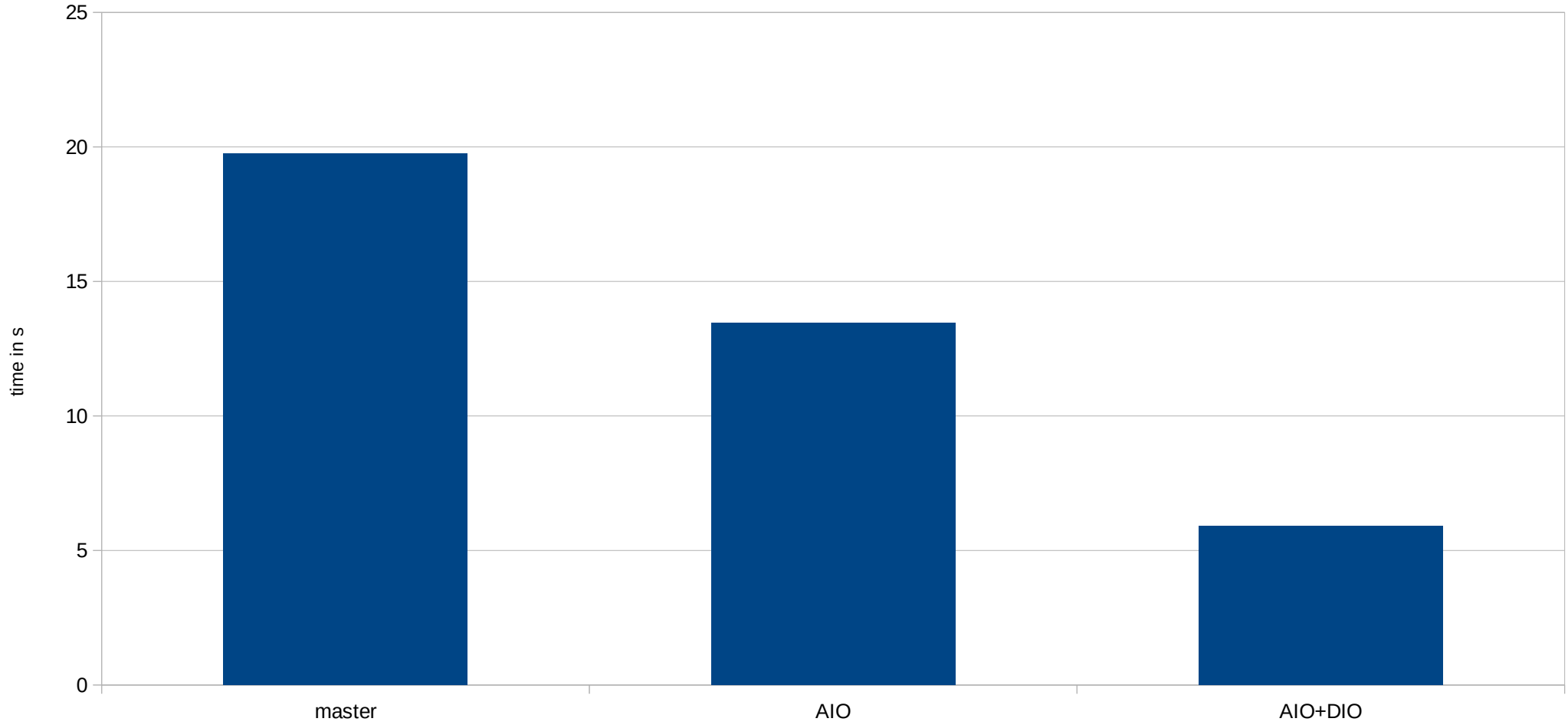
Sequential Scan Performance, Cloud Storage

12GB table, clean OS and PG cache



Sequential read via pg_prewarm

34GB on 2 striped PCIe v3 SSDs

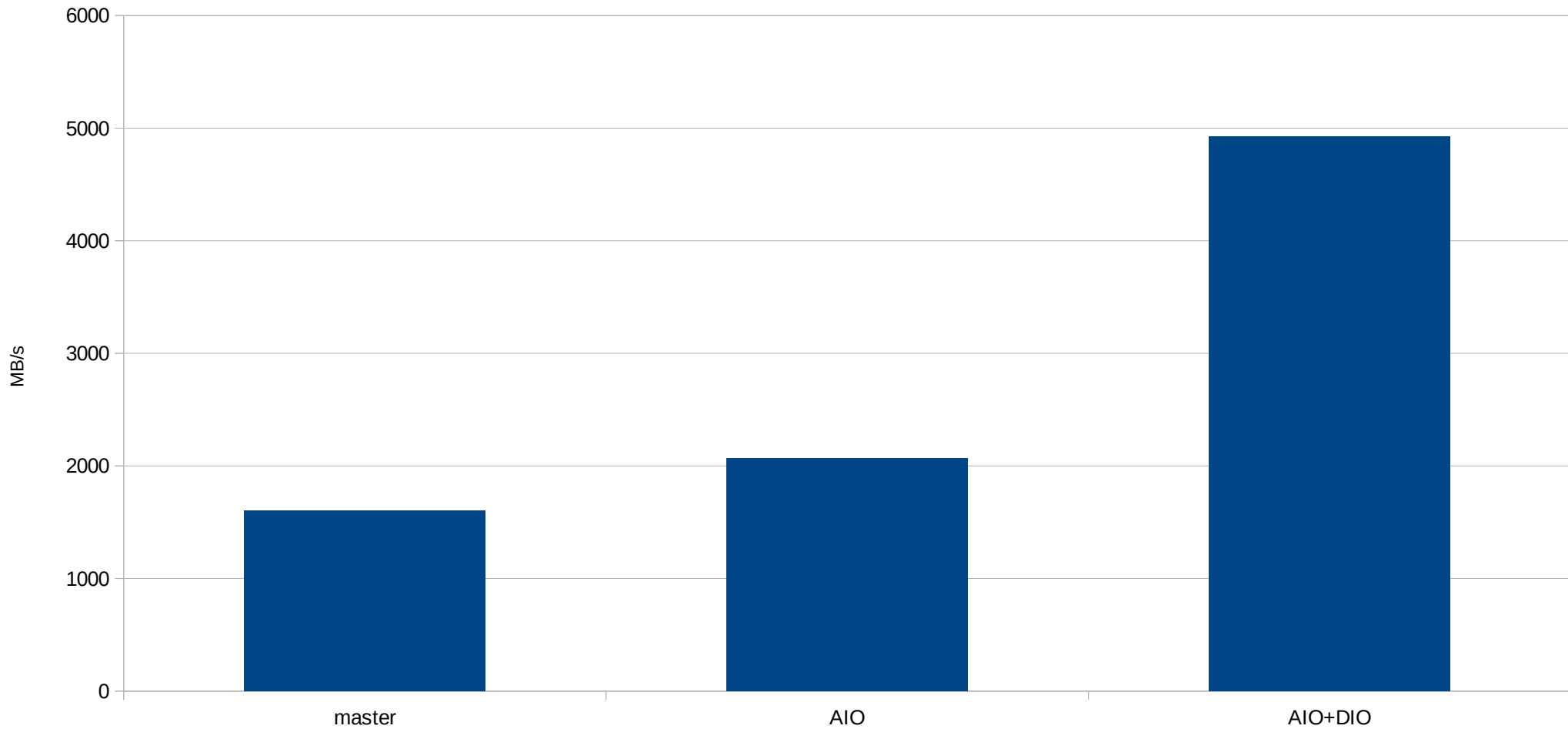


17-18?: checkpointer, bgwriter

- Throughput limited due to CPU overhead
- Limited control over latency impact with buffered IO

checkpoint 35GB of dirty data

stripe of 2 PCIe 3x SSDs, io_uring

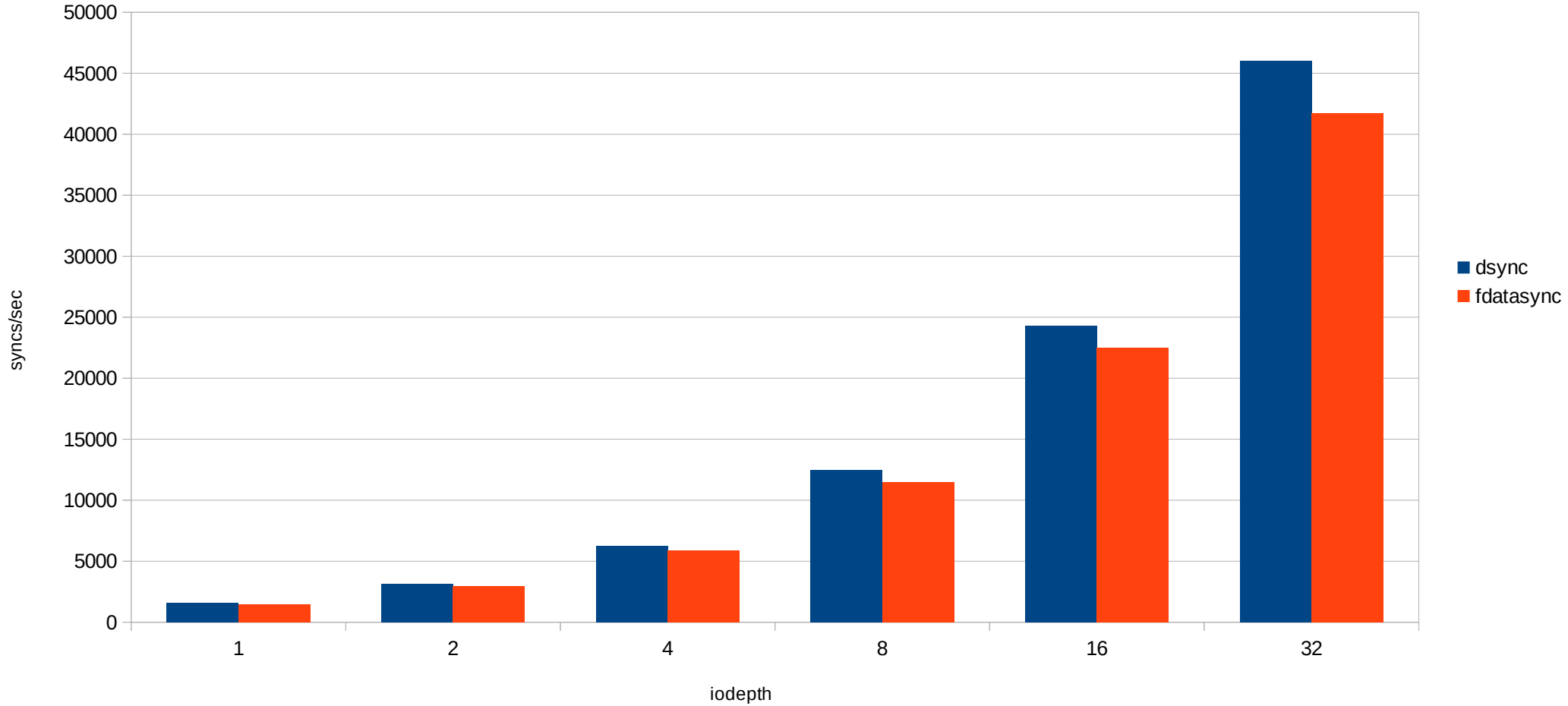


18?: WAL writes

- benefit #1: Do something else during WAL write / flush
 - could get rid of bgwriter (often overloaded, not adaptive)
- benefit #2: Multiple WAL flushes concurrently
 - we have group commit
 - but only one flush in progress
- Hard, gains only very partially realized right now

sync operations/sec, using fio

on cloud storage device with decent latency

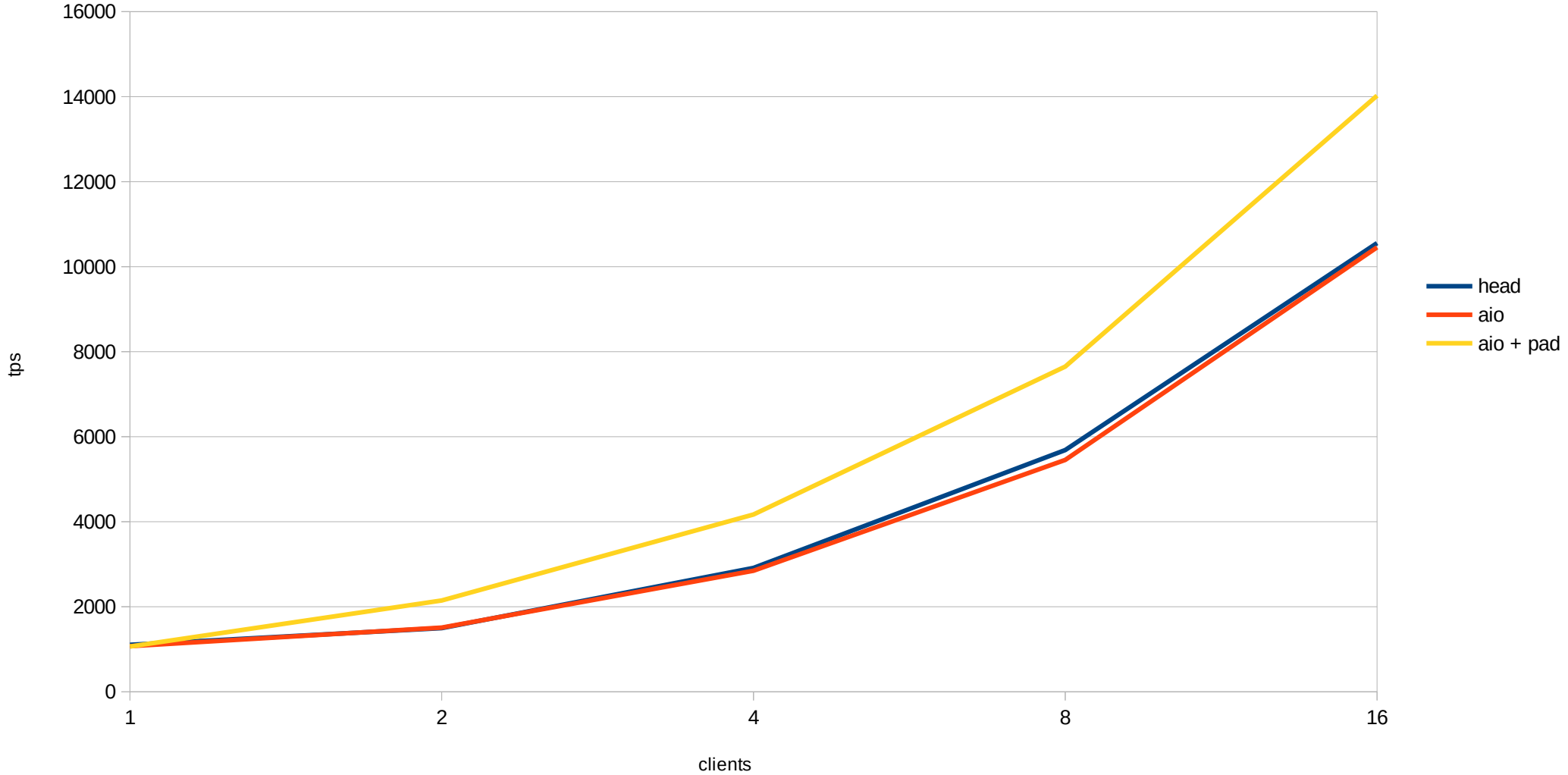


18?: WAL writes

- pgbench transaction: ~450 bytes
- default WAL page size: 8kB
- in-write page cannot be written again before completion
- Problem: to-be-flushed-page is rarely full
- Solution (?): optionally pad partial pages

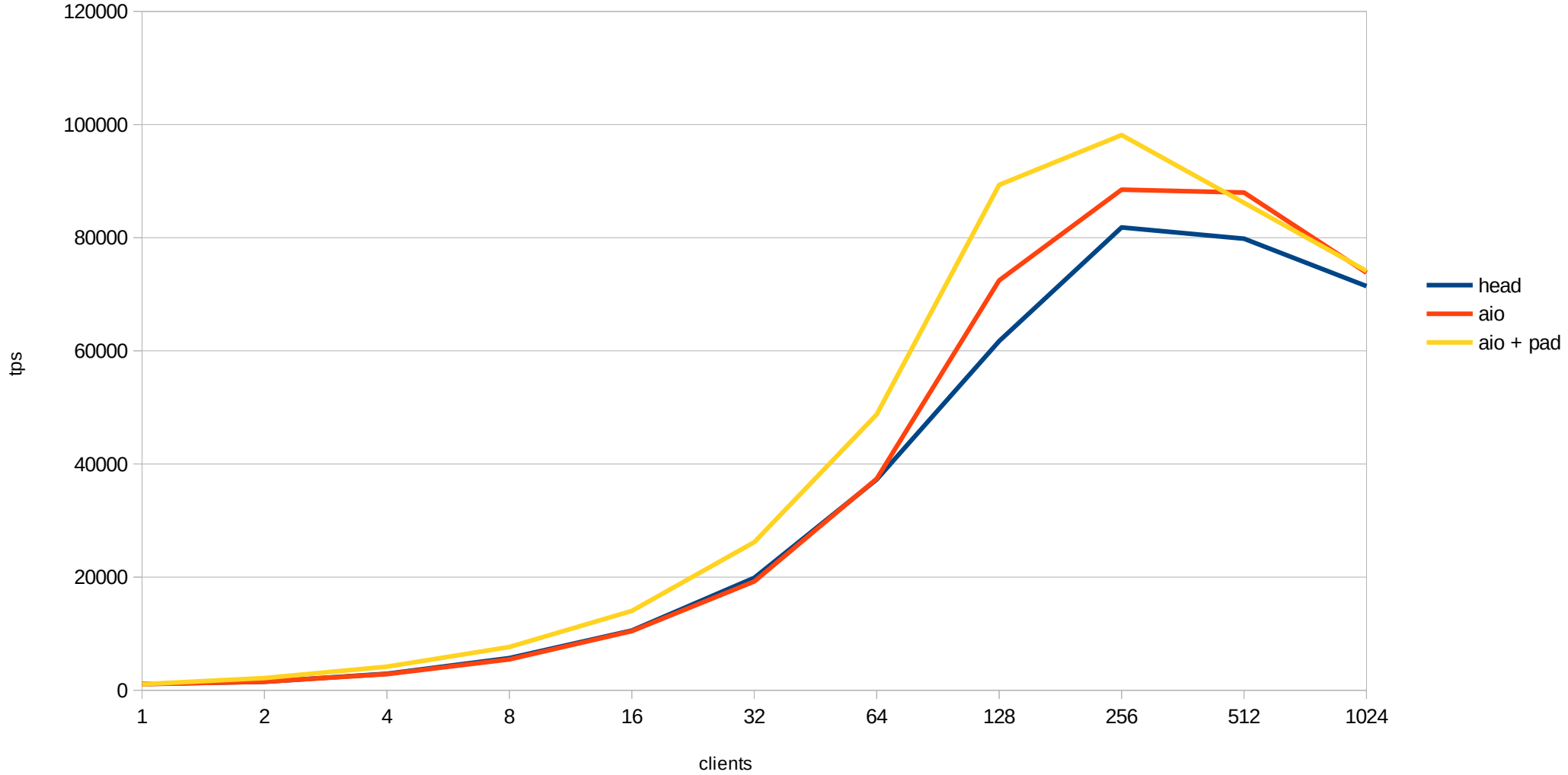
pgbench TPS

full_page_writes=off, scale 2000



pgbench TPS

full_page_writes=off, scale 2000



17-18?: VACUUM

- improved read performance
- improved **write** performance
 - due to asynchronous WAL flushing
 - due to DIO
- better control over latency effects

	high lat cloud disk	lower lat cloud disk
master	94.673 s	33.37 s
aio	12.349 s	7.737 s

17, 18: Other working AIO conversions

- `ProcessSyncRequests()` - checkpoints
- Acquire victim buffers in background
- `SyncDataDirectory()` - startup
- Bitmap heap scans

Potential future AIO users

- various index scans
 - Tomas Vondra is working on some bits
- More vacuuming
- Temp table support
- More everything
- lower-level operations
 - create database
 - vacuum full
 - on startup cleanups
 - filesystem directory iteration

Thanks!

- Colleagues working with me on this
 - David, Melanie, Thomas and others
- Others working on related important pieces
 - Tomas Vondra is working index prefetching
 - Bharath Rupireddy is working feeding walsender from buffers
 - ...
- Heikki Linnakangas has been reviewing parts
- ...
- github.com/anarazel/postgres/tree/aio
- <http://wiki.postgresql.org/wiki/AIO>

The path to using AIO in Postgres

Andres Freund

PostgreSQL Developer & Committer

Email: andres@anarazel.de

Email: andres.freund@microsoft.com

anarazel.de/talks/2023-12-14-pgconf-eu-path-to-aio/path-to-aio.pdf