



1000X WORLD TOUR

13 CITIES | 6 DAYS

Q Why is TigerBeetle



Videos S

tigerbeetle so bad

tigerbeetle so slow

tigerbeetle so expensive

tigerbeetle so good

tigerbeetle so bad reddit

Products More ▾

Tools ▾

actions (OLTP ...



CAPTAIN: TAKE OFF EVERY 'ZIG'!!!



∅ ...

We are so lucky to get a lectures on trade offs from a database company
who's product only stores integers

7:56 AM · Nov 22, 2025



_____.-----'~~~~~:-----'._____
(| : |)
\ ,;,,, : | /
| \%%%\-----:-----/ ~~~~/ |
/ ,\%%%\-----| / @*@ /, \
/_ / `----| ~~~~~ \ _\
(@l) | (@l)
||-----|-----||
||-----|-----||
/_|-----|_\

```
$ tigerbeetle benchmark  
12212 batches in 220.09 s  
transfer batch size = 8189 txs  
transfer batch delay = 0 us  
load accepted = 454365 tx/s  
batch latency p1    = 6 ms  
batch latency p50   = 15 ms  
batch latency p99   = 87 ms  
batch latency p100  = 94 ms
```

```
$ zig/zig build vopr -Drelease -Dvopr-state-machine=testing -- 5787638273934613219
SEED=5787638273934613219

replicas=3
standbys=3
clients=13
request_probability=67/100
idle_on_probability=17/100
idle_off_probability=12/100
one_way_delay_mean=40.584ms ticks
one_way_delay_min=18.342ms ticks
packet_loss_probability=16/100
path_maximum_capacity=3 messages
path_clog_duration_mean=1.407s ticks
path_clog_probability=2/100
packet_replay_probability=5/100
partition_mode=uniform_size
partition_symmetry=symmetric
partition_probability=1/100
unpartition_probability=7/100
partition_stability=200 ticks
unpartition_stability=3 ticks
read_latency_min=26.973ms
```

```
$ cat src/vopr.zig
...
const storage_fault_atlas: Cluster.StorageFaultAtlas.Options = .{
    .faulty_superblock = true,
    .faulty_wal_headers = replica_count > 1,
    .faulty_wal_prepares = replica_count > 1,
    .faulty_client_replies = replica_count > 1,
    // >2 instead of >1 because in R=2, a lagging replica may sync to the leading replica,
    // but then the leading replica may have the only copy of a block in the cluster.
    .faulty_grid = replica_count > 2,
};

...
```

```
$ cat src/vsr/replica.zig
...
    // Disable all dynamic allocation from this point onwards.
    self.static_allocator.transition_from_init_to_static();
...
```

Replica 0

Replica 2

Replica 4

Replica 1

Replica 3

Replica 5

Events:

```
create_account id=1
```

```
create_account id=2
```

```
create_transfer  
debit_account_id=1  
credit_account_id=2  
amount=100
```

```
create_transfer  
debit_account_id=1  
credit_account_id=2  
amount=150
```

```
create_transfer  
debit_account_id=1  
credit_account_id=2  
amount=30
```

----- State Machine ----->

State:

```
account id=1  
debit_amount=280  
credit_amount=0
```

```
account id=2  
debit_amount=0  
credit_amount=280
```

```
class Account {
    debit_balance: number
    credit_balance: number
}

class Transfer {
    debit_account_id: number
    credit_account_id: number
    amount: number
}

class State {
    accounts: Map<number, Account>
    transfers: Array<Transfer>
}

// Mutates State in-place :)
function create_transfer(state: State, transfer: Transfer): State {
    state.transfers.append(transfer);
    state.accounts.get(transfer.debit_account_id).debit_balance += transfer.amount
    state.accounts.get(transfer.credit_account_id).credit_balance += transfer.amount
}
```

```
class Account {
    debit_balance: number
    credit_balance: number
}

class Transfer {
    debit_account_id: number
    credit_account_id: number
    amount: number
}

class State {
    accounts: Map<number, Account>
    transfers: Array<Transfer>
}

// Mutates State in-place :)
function create_transfer(state: State, transfer: Transfer): State {
    state.transfers.append(transfer);
    state.accounts.get(transfer.debit_account_id).debit_balance += transfer.amount
    state.accounts.get(transfer.credit_account_id).credit_balance += transfer.amount
}
```

Events:

```
create_account_v0.1 id=1
```

```
create_account_v0.1 id=2
```

```
create_transfer_v0.2  
debit_account_id=1  
credit_account_id=2  
amount=100
```

```
create_transfer_v0.2  
debit_account_id=1  
credit_account_id=2  
amount=150
```

```
create_transfer_v0.2  
debit_account_id=1  
credit_account_id=2  
amount=30
```

----- State Machine v0.1 ----->

State:

```
account id=1  
debit_amount=280  
credit_amount=0
```

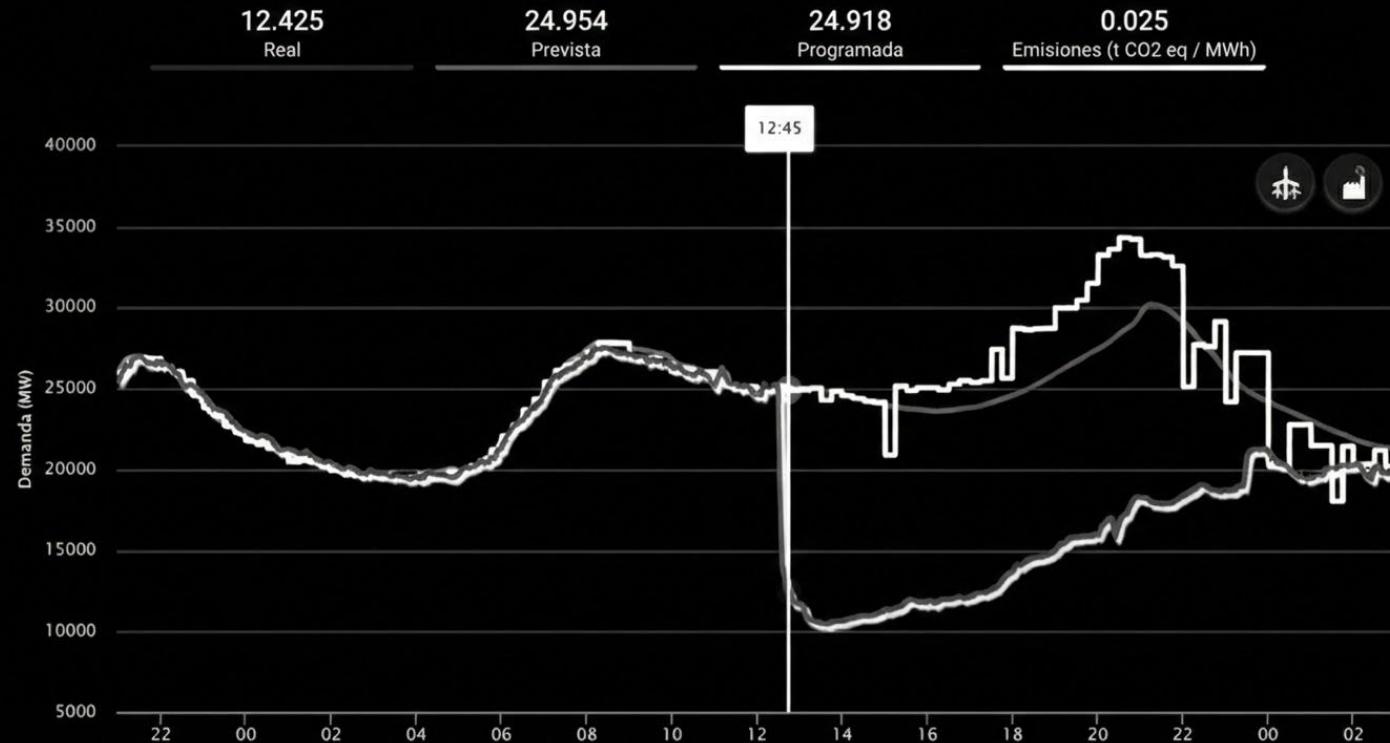
```
account id=2  
debit_amount=0  
credit_amount=280
```

->

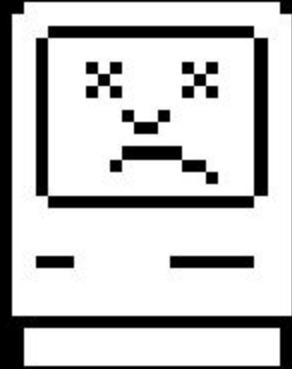
```
$ ssh root@machine-0  
machine-0$ systemctl stop tigerbeetle
```

...

```
$ ssh root@machine-5  
machine-5$ systemctl stop tigerbeetle
```


```
# dmesg
kernel: amdgpu 0000:06:00.0: [drm] *ERROR* dc_dmub_srv_log_diagnostic_data: DMCUB error
kernel: amdgpu 0000:06:00.0: [drm] *ERROR* [CRTC:79:crtc-0] flip_done timed out
kernel: [drm:amdgpu_dm_atomic_check [amdgpu]] *ERROR* [CRTC:79:crtc-0] hw_done or flip_done timed out
kernel: amdgpu 0000:06:00.0: [drm] *ERROR* flip_done timed out
kernel: amdgpu 0000:06:00.0: [drm] *ERROR* [CRTC:79:crtc-0] commit wait timed out
```



Events:

```
create_account_v0.1 id=1  
create_account_v0.1 id=2  
upgrade from=v0.1 to=v0.2  
create_transfer_v0.2  
  debit_account_id=1  
  credit_account_id=2  
  amount=100  
  
create_transfer_v0.2  
  debit_account_id=1  
  credit_account_id=2  
  amount=150  
  
create_transfer_v0.2  
  debit_account_id=1  
  credit_account_id=2  
  amount=30
```

----- State Machine v0.1 ----->

->

State:

```
account id=1  
  debit_amount=280  
  credit_amount=0  
  
account id=2  
  debit_amount=0  
  credit_amount=280
```

----- State Machine v0.2 ----->

```
$ curl -Lo tigerbeetle.zip https://linux.tigerbeetle.com && unzip tigerbeetle.zip
Archive: tigerbeetle.zip
inflating: tigerbeetle
```

```
$ curl -Lo tigerbeetle.zip https://mac.tigerbeetle.com && unzip tigerbeetle.zip
Archive: tigerbeetle.zip
inflating: tigerbeetle
```

```
> curl.exe -Lo tigerbeetle.zip https://windows.tigerbeetle.com; Expand-Archive tigerbeetle.zip .
Archive: tigerbeetle.zip
inflating: tigerbeetle.exe
```

```
$ binwalk tigerbeetle  
/home/user/tigerbeetle
```

DECIMAL	HEXADECIMAL	DESCRIPTION
0	0x0	ELF binary, 64-bit executable
4092789	0x3E7375	ELF binary, 64-bit executable
8161925	0x7C8A85	ELF binary, 64-bit executable
12265853	0xBB297D	ELF binary, 64-bit executable
16367621	0xF9C005	ELF binary, 64-bit executable

```
Analyzed 1 file for 85 file signatures (187 magic patterns) in 126.0 milliseconds
```

```
# uname -a
Linux laptop 6.16.9-arch1-1 #1 SMP PREEMPT_DYNAMIC Thu, 25 Sep 2025 17:17:24 +0000 x86_64 GNU/Linux
```

```
# uname -a
Darwin MacBook-Pro 23.5.0 Darwin Kernel Version 23.5.0:
Fri Mar 15 00:23:12 PST 2024; root:xnu-10002.121.3~2/RELEASE_ARM64_T6000 arm64
```

```
# ver
Microsoft Windows [Version 10.0.22631.3593]
```

```
$ clamscan -r /home/user/downloads/  
  
/home/user/downloads/tigerbeetle.exe: Win.Trojan.Generic-1234567 FOUND  
  
----- SCAN SUMMARY -----  
Known viruses: 8723641  
Engine version: 1.2.1  
Scanned directories: 12  
Scanned files: 57  
Infected files: 2  
Data scanned: 22.11 MB  
Data read: 23.80 MB (ratio 0.93:1)  
Time: 9.337 sec (0 m 9 s)
```

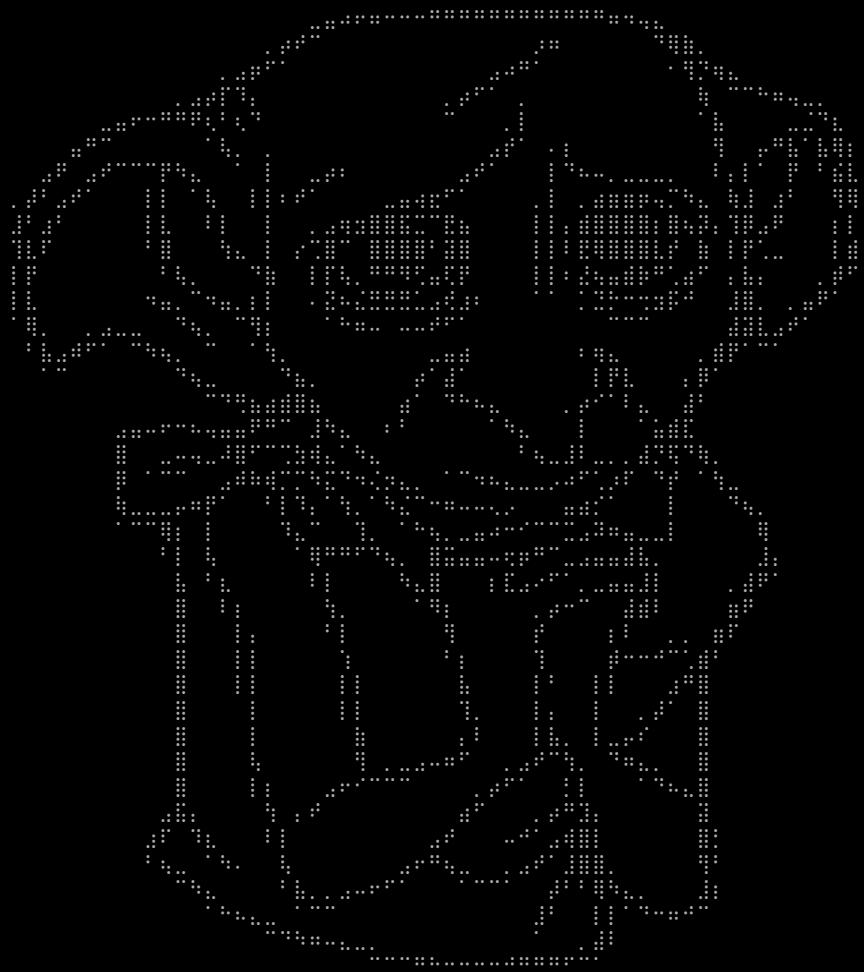
```
$ ldd tigerbeetle-launcher
 linux-vdso.so.1 (0x0000740410e2f000)
 tigerbeetle.so.40 => /usr/lib/tigerbeetle.so.40 (0x0000740410dbb000)
 tigerbeetle.so.39 => /usr/lib/tigerbeetle.so.39 (0x0000740410dbb000)
```

```
const tigerbeetle_40_binary = @embedFile("tigerbeetle-40");

...

try shell.cwd.writeFile(.{
    .sub_path = "/tmp/tigerbeetle_40",
    .data = tigerbeetle_40_binary,
});
```

```
// Use objcopy to add in our new body, as well as its header - even though the
// header is still zero!
try shell.exec(
    \\"{llvm_objcopy} --enable-deterministic-archives --keep-undefined
    \\
    \\--add-section .tb_mvb={body}
    \\--set-section-flags .tb_mvb=contents,noload,readonly
    \\
    \\--add-section .tb_mvh={header_zero}
    \\--set-section-flags .tb_mvh=contents,noload,readonly
    \\
    \\{working}
, .{
    .llvm_objcopy = options.llvm_objcopy,
    .body = sections.body,
    .header_zero = sections.header_zero,
    .working = tigerbeetle_working,
});
```



```
$ xxd any-modern.exe
00000000: 4d5a 7800 0100 0000 0400 0000 0000 0000 0000 0000 0000 0000 MZx.....
00000014: 0000 0000 4000 0000 0000 0000 0000 0000 0000 0000 0000 0000 ....@.....
00000028: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 .....
0000003c: 7800 0000 0e1f ba0e 00b4 09cd 21b8 014c cd21 5468 x.....!..L.!Th
00000050: 6973 2070 726f 6772 616d 2063 616e 6e6f 7420 6265 is program cannot be
00000064: 2072 756e 2069 6e20 444f 5320 6d6f 6465 2e24 0000 run in DOS mode.$..
00000078: 5045 0000 6486 0800 04ad b4c6 0000 0000 0000 0000 PE..d.....
```

> Load commands

>> Segment load command

>>> Segment number and section numbers

>> Link libraries

>>> Link library ordinal numbers

>> __LINKEDIT symbol table

>>> Symbol table organization

>> __LINKEDIT Symbol table information

>>> Indirect table

>> __LINKEDIT Compressed table

>>> Binding information

>> Application main entry point

>> Application UUID number

>> Minimum OS version



Mach-O Universal Header

Offset	Bytes	Description
0	4	Magic Number
4	4	Number of binaries

Mach-O Universal File Entries

Offset	Bytes	Description
0	4	CPU Type
4	4	CPU Subtype
8	4	File Offset
12	4	Size
16	4	Alignment



```
$ file tigerbeetle
tigerbeetle: Mach-O universal binary with 6 architectures:
  [arm64:\012- Mach-O 64-bit arm64 executable,
flags:<NOUNDEFs|DYLDLINK|TWOLEVEL|NO_REEXPORTED_DYLIBS|PIE|HAS_TLV_DESCRIPTOR
S>]

  [\012- x86_64:\012- Mach-O 64-bit x86_64 executable,
flags:<NOUNDEFs|DYLDLINK|TWOLEVEL|WEAK_DEFINES|BINDS_TO_WEAK|NO_REEXPORTED_DY
LIBS|PIE|HAS_TLV_DESCRIPTORs>]

  [\012- vax]
  [\012- romp]
  [\012- ns32032]
  [\012- ns32332]
```



```
$ man proxytunnel
```

```
...
```

BUGS

This software is bug-free, at least we'd like to think so. If you do not agree with us,
please provide the proof with your friendly report at
<https://github.com/proxytunnel/proxytunnel/issues> :)

```
...
```

```
$ man 3 read
```

SYNOPSIS

```
ssize_t read(int fildes, void *buf, size_t nbytes);
```

The value returned may be less than nbytes if the number of bytes left in the file is less than nbytes, if the read() request was interrupted by a signal, or if the file is a pipe or FIFO or special file and has fewer than nbytes bytes immediately available for reading.

```
$ cat main.zig
pub fn main() !void {
    _ = multiply(128, 2);
}

pub fn multiply(a: u8, b: u8) u8 {
    return a * b;
}

$ zig run main.zig
thread 464530 panic: integer overflow
/tmp/main.zig:6:14: 0x113e8b4 in multiply (main.zig)
    return a * b;
               ^
/tmp/main.zig:2:16: 0x113d370 in main (main.zig)
    _ = multiply(128, 2);
               ^
```

```
$ cat ~/lessons
* drop down a level of abstraction
* use existing guarantees
* problems lie at the edges, always
* simplicity for the user
```

