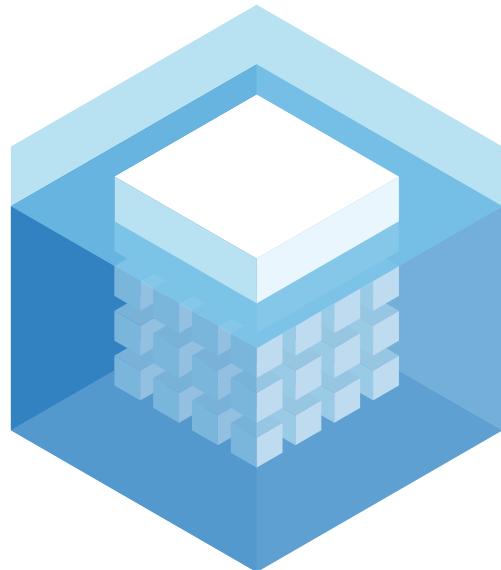


Zarr in Production Using Icechunk

Sebastian Galkin



 earthmover

The setup

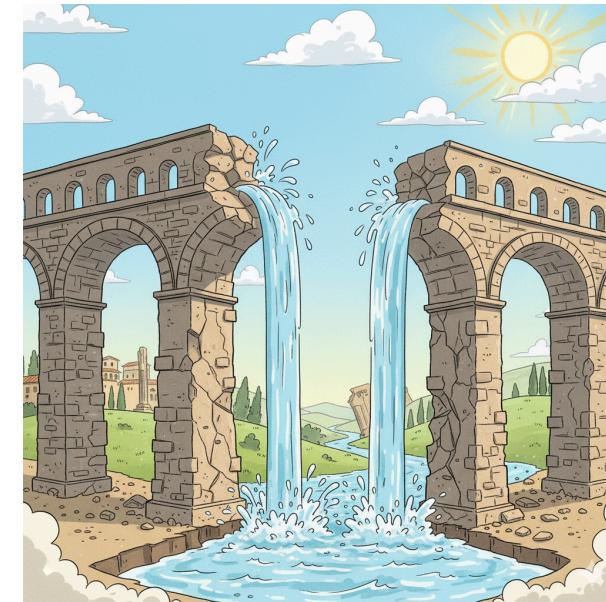
```
NUM_ROWS = 1_000
NUM_COLS = 2

def create_array(store: StoreLike) -> None:
    zarr.create_array(
        store=store, chunks=(1, 1), dtype=np.int8,
        data=np.zeros((NUM_ROWS, NUM_COLS))
    )

def mean(store: StoreLike) -> None:
    array = zarr.open_array(store)
    col = random.randint(0, NUM_COLS - 1)
    m = np.mean(array[:, col])
    print(f"Mean is {m}")
```

The problem of interrupted updates

```
array = zarr.open_array(store)
for x in range(NUM_ROWS):
    for y in range(NUM_COLS):
        array[x, y] = 1
        if random.random() < probability:
            raise ValueError("ingest broke")
```



The problem of interrupted updates

```
array = zarr.open_array(store)
for x in range(NUM_ROWS):
    for y in range(NUM_COLS):
        array[x, y] = 1
        if random.random() < probability:
            raise ValueError("ingest broke")
```

- Mean is 0.385
- Mean is 0.115
- Mean is 0.997
- Mean is 1.0



The problem with concurrent readers

```
def update(store: StoreLike) -> None:  
    array = zarr.open_array(store)  
    # Update every element to 1  
    array[:] = 1
```



```
while True:  
    mean(store)
```



The problem with concurrent readers

```
def update(store: StoreLike) -> None:  
    array = zarr.open_array(store)  
    # Update every element to 1  
    array[:] = 1
```

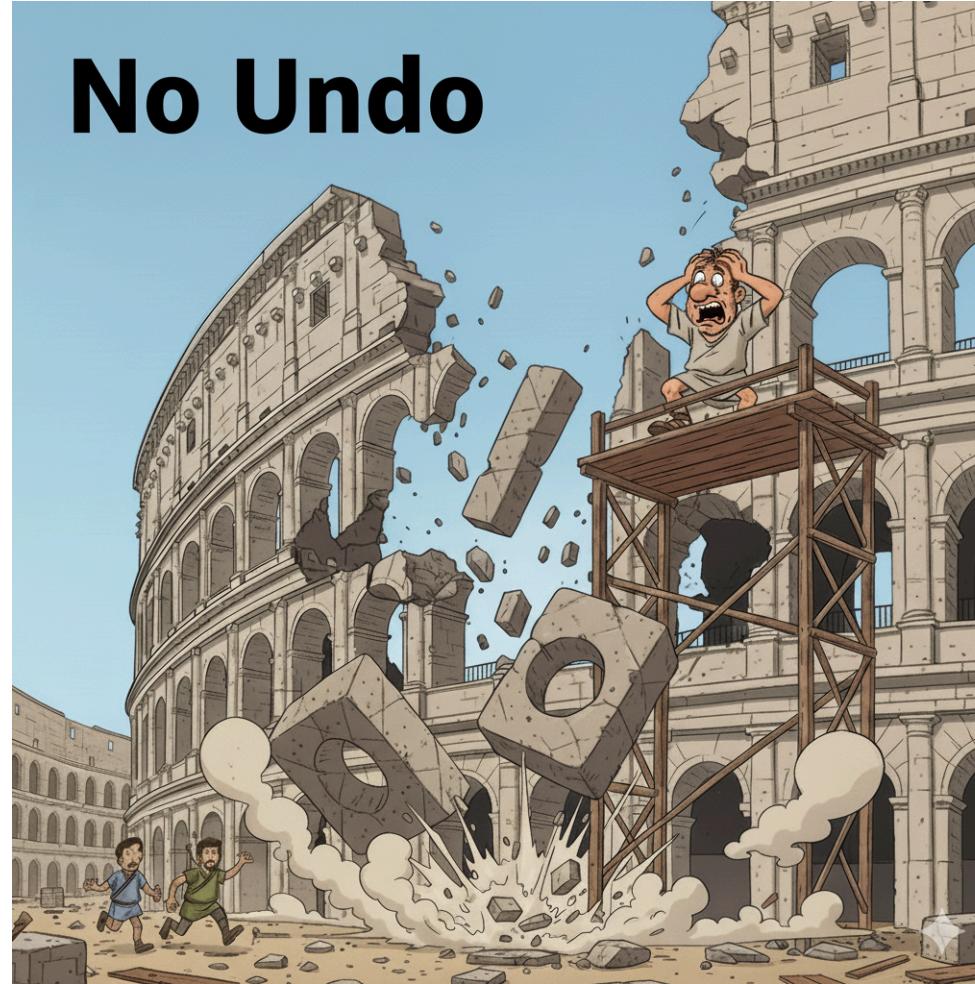


```
while True:  
    mean(store)
```

- Mean is 0.0
- Mean is 0.0
- **Mean is 0.459**
- Mean is 1.0
- Mean is 1.0



No Undo



Icechunk = Zarr on ACID

- Atomic
- Consistent
- Isolated
- Durable



ACID can solve most problems

```
def update(repo: ic.Repository) -> None:
    with repo.transaction("main", message="Update done") as store:
        array = zarr.open_array(store)
        array[:] = 1

while True:
    session = repo.readonly_session(branch="main")
    mean(session.store)
```

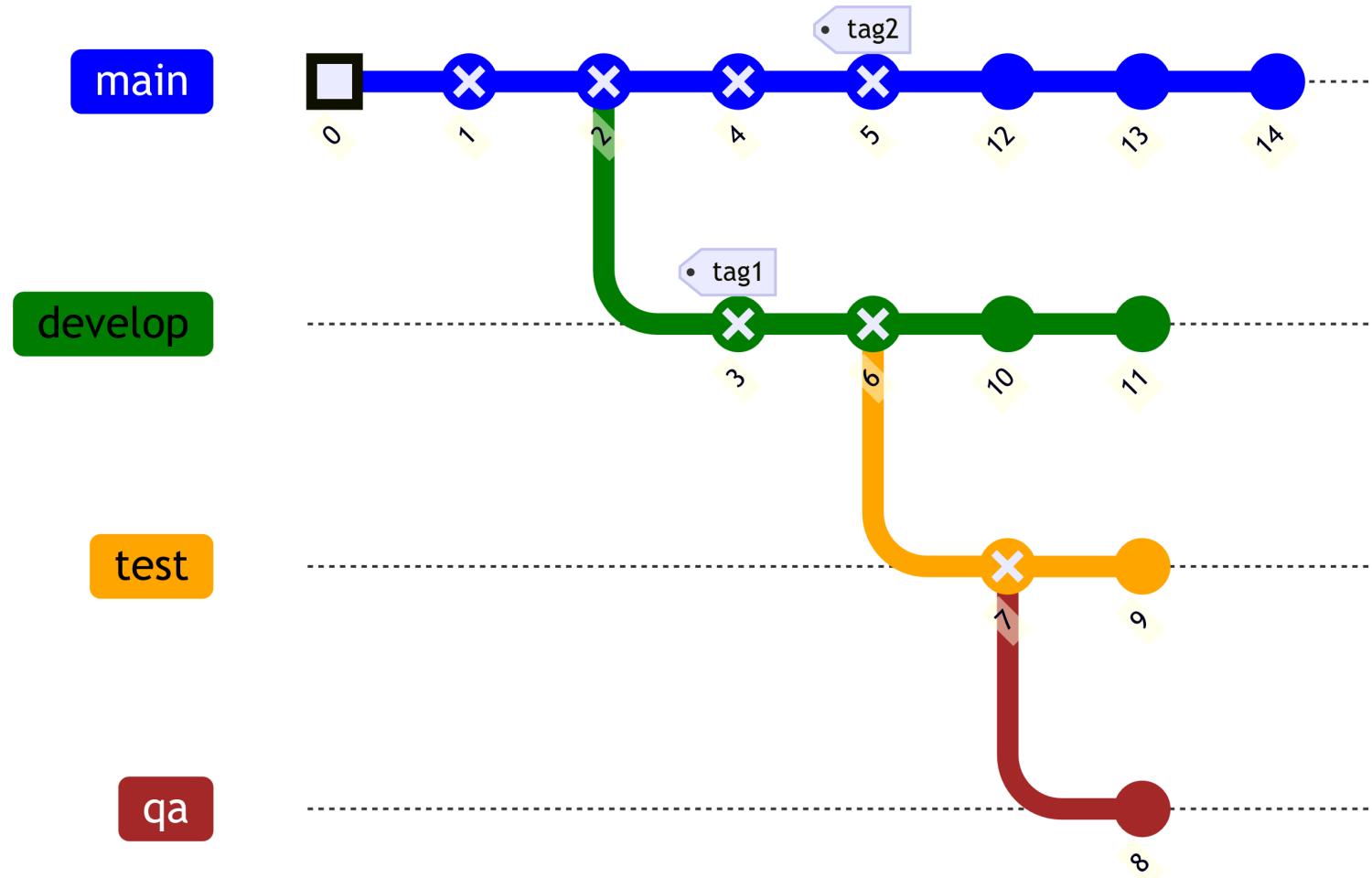
ACID can solve most problems

```
def update(repo: ic.Repository) -> None:  
    with repo.transaction("main", message="Update done") as store:  
        array = zarr.open_array(store)  
        array[:] = 1  
  
while True:  
    session = repo.readonly_session(branch="main")  
    mean(session.store)
```

- Mean is 0.0
- Mean is 0.0
- Mean is 1.0

Updates become **atomic**, **consistent**, and **isolated**

Solving the Undo problem



Fully Zarr, Xarray, and Dask compatible

```
import xarray as xr

ds = ...

# Start an icechunk session
session = repo.writable_session(branch = "test-xarray")

# write the data to zarr
ds.to_zarr(session.store, group='my_group')

# commit the data
session.commit("My first commit 😊")
```

Simple but powerful API

```
# Git-like operations
repo.ancestry(...)

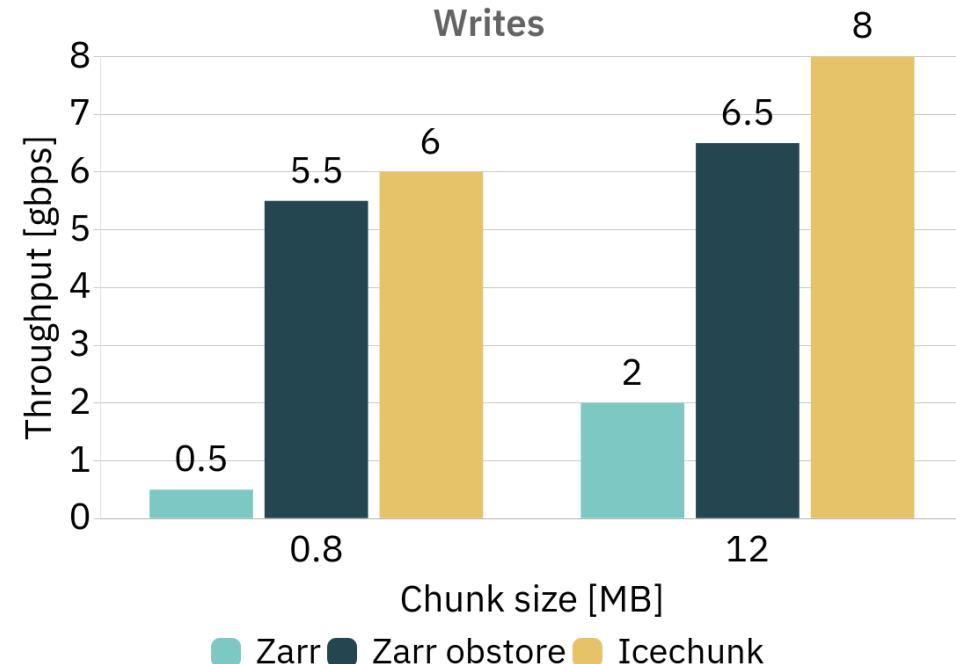
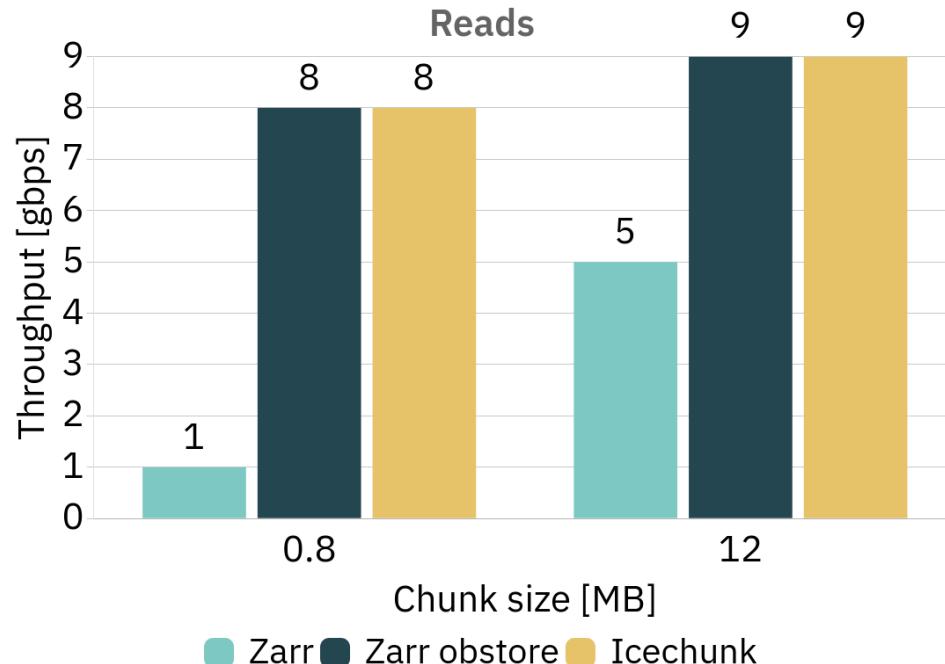
repo.create_branch(...)
repo.create_tag(...)
repo.lookup_branch(...)
repo.list_branches()

# Powerful features
repo.diff(
    from_branch="main",
    to_tag="v1.0"
)

# Conflict detection
Session.rebase(solver: ConflictSolver)
```

Icechunk is fast out of the box

Icechunk is fast out of the box



Icechunk is production grade

- Released 1 year ago
- Open format specification
- pip install icechunk==1.1.9
- Supported in zarr-python and zarrs
- Any object store (S3, GCS, R2, Azure blob, Tigris, MinIO, etc.)
- Lots of organizations are using Icechunk for their production data
- Fully open source, Apache 2.0 licensed

Questions

You can start using Icechunk today

- <https://icechunk.io>

For help:

- Introductory workshop tomorrow at 9am
- Ask questions to the Earthmover team today or tomorrow
- Play with Icechunk public datasets in Arraylake: <https://earthmover.io/>
- Go to <https://icechunk.io> and join the Icechunk community slack
- Earthmover's blog