Managing Multiple Sources of Truth in Distributed Systems

Adam Wolfe Gordon
DigitalOcean
What is a Source of Truth?

- A component that stores state.

For example:

- A database or key/value store.
- An external auth system storing user details.
- A hypervisor running virtual machines.
- A distributed storage system.
Multiple Sources of Truth

- Multiple systems storing state.
- Specifically, state about the same objects.
- **Not** caches.
Consistency

The main problem with having multiple sources of truth is keeping them consistent.
Motivating Example

Storage Volume Management

- Ceph - Underlying storage.
- Consul - KV store for metadata.
- Billing - Generates invoices.
- QEMU - VMs using storage.
Inconsistencies

- A single user operation involves multiple systems.
- Systems can fail at any time!
Strategies
Strategy 1
Avoid Multiple Sources of Truth

● Whenever possible, keep state in one place.
Strategy 2
Exposé Only One Source of Truth

- Pick a source of truth to expose to users.
- This limits visibility of any inconsistencies.

For example:

In our volume management system, all requests return data from Consul.
Strategy 3
Order Operations Carefully

- When creating state, create in the primary source of truth **last**.
- When updating state, update the primary source of truth **last**.
- When deleting state, delete from the primary source of truth **first**.
Strategy 4
Deferred Work
Record Intents

- Record the **intent** of the user on each request.
- Fail immediately if recording the intent fails.

For example:

In our volume management system, we record intents in Consul.
Process Intents Out-of-Context

- Process intents in the background.
- Check all sources of truth for consistency.
- Fix inconsistencies when it’s safe to do so.
Intent Example: Volume Creation

User

Create volume

Volume Management Services

Add creation intent
OK
Create volume
OK
Add volume metadata
OK

Ceph Consul

Intent Processor

Billing

Get intents
<Intents>

Creation intent handler

Get volume metadata
<Metadata> OR Not Found

alt [Metadata found]
Volume created
OK

[Metadata not found]
Get volume info
<Volume> OR Not Found

alt [Volume exists]
Log that there is an orphaned volume.
Intent Example: Volume Resize

User

Resize volume

OK

Volume Management Services

Ceph Consul

Intent Processor

Billing

User

Add resize intent

OK

Resize volume

OK

Update volume metadata

OK

Get intents

<Intents>

Resize intent handler

Get volume metadata

<Metadata>

Get volume info

<Volume>

alt [Size inconsistent]

Update volume metadata

OK

alt [Size updated]

Volume resized

OK
Intent Example: Volume Deletion

User

Volume Management Services

Delete volume

Add deletion intent

OK

Mark volume metadata deleted

OK

Ceph Consul

Intent Processor

Get intents

<Intents>

Deletion intent handler

Get volume metadata

<Metadata>

alt [Metadata marked deleted]

Delete volume

OK

Volume deleted

Delete volume metadata

OK

Billing
Intent Processor Recordkeeping

- Mark intents as claimed before processing.
- At start, process marked intent first.
- Delete intents once they’re fully processed.
Gotchas

- Actions can be dependent on one another.
- Things may change before intent processing.
- The intent processor can also fail.
- Intents may be processed too early.
Strategy 5
Scrubbing
The Basics of Scrubbing

- Examine each object in the system.
- Interrogate all sources of truth.
- Surface any inconsistencies that are found.
- Potentially fix inconsistencies.
Surfacing Inconsistencies

- **Logging** - Record what’s inconsistent.
- **Metrics** - Count inconsistencies.
- **Alerting** - Make sure someone knows.
Gotchas

- Scrubbing and intent processing can overlap.
- Choosing scrub frequency is tricky.
In Summary
Five Strategies

1. **Avoid** multiple sources of truth.
2. **Limit** visibility of inconsistencies by choosing a primary source of truth.
3. **Minimize** inconsistencies by ordering operations carefully.
4. **Fix** inconsistencies with deferred intent processing.
5. **Detect** inconsistencies by scrubbing.
Thank You!

Adam Wolfe Gordon
awg@do.co