

Arvados - Bug #16665

Keepproxy reports error 413 (entity too large) even if the original error was something else

08/04/2020 08:55 PM - Lucas Di Pentima

Status:	Resolved	Start date:	07/22/2021
Priority:	Normal	Due date:	
Assigned To:	Lucas Di Pentima	% Done:	100%
Category:	Keep	Estimated time:	0.00 hour
Target version:	2021-08-04 sprint		
Description			
This makes debugging really annoying.			
Subtasks:			
Task # 17930: Review 16665-keepproxy-spurious-413-status			Resolved

Associated revisions

Revision 68b9c7d3 - 07/27/2021 02:35 PM - Lucas Di Pentima

Merge branch '16665-keepproxy-spurious-413-status' into main. Closes #16665

Arvados-DCO-1.1-Signed-off-by: Lucas Di Pentima <lucas.dipentima@curii.com>

History

#1 - 06/23/2021 06:09 PM - Lucas Di Pentima

Maybe related to this: go-staticcheck (via vscode) is showing me a warning that says:

unreachable case clause: git.arvados.org/arvados.git/sdk/go/keepclient.OversizeBlockError will always match before git.arvados.org/arvados.git/sdk/go/keepclient.InsufficientReplicasError (SA4020)

...for the case keepclient.InsufficientReplicasError: in keepproxy.go's Put() function. Both types are aliases of error so it seems the compiler won't take them as different.

#2 - 06/23/2021 06:13 PM - Peter Amstutz

```
type InsufficientReplicasError error
```

```
type OversizeBlockError error
```

```
switch err.(type) {
case nil:
    status = http.StatusOK
    _, err = io.WriteString(resp, locatorOut)

case keepclient.OversizeBlockError:
    // Too much data
    status = http.StatusRequestEntityTooLarge

case keepclient.InsufficientReplicasError:
}
```

I think this is an inconsistency in Go. If you declare a type that is the same as another type, at compile time they will act like different types, but at runtime they will be the same type.

Does this work?

```
type InsufficientReplicasError struct { error }
```

#3 - 06/23/2021 06:14 PM - Peter Amstutz

- Target version changed from Arvados Future Sprints to 2021-07-21 sprint

#4 - 07/07/2021 04:01 PM - Peter Amstutz

- Target version changed from 2021-07-21 sprint to 2021-08-04 sprint

#5 - 07/21/2021 03:31 PM - Peter Amstutz

- Assigned To set to Lucas Di Pentima

#6 - 07/22/2021 06:38 PM - Lucas Di Pentima

- Status changed from New to In Progress

#7 - 07/22/2021 09:04 PM - Lucas Di Pentima

Updates at [e10c23d41](#) - branch 16665-keepproxy-spurious-413-status

Test run: <https://ci.arvados.org/job/developer-run-tests/2606/>

- Following Peter's suggestion, updates InsufficientReplicasError & OversizeBlockError error types by wrapping error in a struct, so that the former isn't shadowed by the latter.
- Updates tests.

#8 - 07/27/2021 02:11 PM - Tom Clegg

LGTM, thanks.

BTW, about the language issue: error is an interface type, therefore type FooError error defines an interface type, therefore anything that implements the error interface { Error() string } has to match FooError in a type switch. After this fix, type FooError struct { error } is a concrete type, so FooError in a type switch is unambiguous. I think it would have been more normal to use var ErrFoo = errors.New(...) and if err == ErrFoo { ... } for these... oh well.

#9 - 07/27/2021 03:17 PM - Lucas Di Pentima

- % Done changed from 0 to 100

- Status changed from In Progress to Resolved

Applied in changeset [arvados|68b9c7d30c7f47f8a9f9cff8a327fa9a3812d4da](#).