

Arvados - Bug #7492

[keepproxy] Report upstream server errors as 502 Bad Gateway instead of 404

10/08/2015 07:19 PM - Peter Amstutz

Status:	Resolved	Start date:	10/14/2015
Priority:	Normal	Due date:	
Assigned To:	Radhika Chippada	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	2015-10-28 sprint		

Description

Keepproxy (which uses keepclient) is failing requests because one of the upstream servers returned "read tcp 10.26.0.7:25107: use of closed network connection". This is being returned to the client as a 404, but actually should be a 502 (Bad Gateway) which would trigger retry logic on the client.

Requirements

If an error occurs in (*keepclient.KeepClient)Get() then keepproxy's http response status must be:

- 404 if every keepstore service returned 404
- 422 if every keepstore service returned a non-retryable error (but some didn't return 404)
- 502 if any keepstore service had still returned a retryable error when tries_remaining ran out

Also, the list of errors received from the upstream keepstore services should be propagated to keepproxy's client in the response body. (Currently this information is logged by the SDK, so it shows up in keepproxy's log file but it is not available to the originator of the request.)

Implementation

We need to update (*keepclient.KeepClient)Get() to make it *possible* for keepproxy to distinguish between the above outcomes.

keepclient:

- create an Error interface (similar to <https://golang.org/pkg/net/#Error>)

```
◦ type Error struct {  
    error  
    Temporary() bool // Is the error temporary?  
}
```

- create a multipleResponseError type that implements Error

```
◦ type multipleResponseError struct {  
    error  
    isTemp bool  
}  
func (e *multipleResponseError) Temporary() bool {  
    return e.isTemp  
}
```

- create a NotFoundError type that implements Error

```
◦ type NotFoundError struct {  
    multipleResponseError  
}
```

- Change BlockNotFound to a NotFoundError

```
◦ var BlockNotFound = &NotFoundError{multipleResponseError{
```

```
    error: errors.New("Block not found"),
    isTemp: false,
  }}
}
```

- in Get(), count how many responses are http.StatusNotFound
- when returning from Get() after all attempts have failed, make a NotFoundError or a multipleResponseError:

```
◦ if count404 == len(serversToTry) {
    err = BlockNotFound
  } else {
    err = &multipleResponseError{
      error:    fmt.Errorf("%s %s failed: %v", method, locator, errs),
      isTemp: len(serversToTry) > 0,
    }
  }
}
```

(Defer: add a way for clients to see a slice of all errors encountered, or at least the last error from each server. For now, that information is only available as a string, via the usual Error() method.)

keepproxy:

- Do a type select on the error returned from keepclient.Get() to determine the appropriate HTTP response

```
◦ switch err := err.(type) {
case keepclient.NotFoundError:
    // respond 404
case keepclient.Error:
    if err.Temporary() {
        // respond 502
    } else {
        // respond 422
    }
default:
    // respond 500
}
```

- In all cases, return err.Error() in the response body.

Subtasks:

Task # 7557: Review branch: 7492-keepproxy-upstream-errors

Resolved

Related issues:

Related to Arvados - Bug #7491: [SDK] Go keepclient should retry on network e...

Resolved

10/08/2015

Associated revisions

Revision b90d3ea4 - 10/28/2015 04:22 PM - Radhika Chippada

closes #7492

Merge branch '7492-keepproxy-upstream-errors'

History

#1 - 10/08/2015 07:20 PM - Peter Amstutz

- Description updated

#2 - 10/08/2015 08:12 PM - Nico César

same as in <https://dev.arvados.org/issues/7491#note-4>

#3 - 10/14/2015 07:38 PM - Brett Smith

- Target version set to 2015-10-28 sprint

#4 - 10/14/2015 07:45 PM - Radhika Chippada

- Assigned To set to Radhika Chippada

#5 - 10/19/2015 01:43 PM - Peter Amstutz

keepclient.go:144 func getOrHead()

If the client was unable to get a final answer from one or more servers (either due to network failure or the server returned a HTTP 408, 429, or 500+ error) it should not return BlockNotFound, it should return some other error. Keepproxy probably does not need to be changed, because it already responds 502 Gateway error as the default case for errors other than BlockNotFound.

Comments from Tom on another [#7491](#):

- Return BlockNotFound only if all servers succeeded and replied exactly 404.
- If not returning BlockNotFound, return something like `fmt.Errorf("%+v", errs)` so the caller can see/report what went wrong.

Unresolved questions (ask Tom):

- What about 401 (Unauthorized) and 403 (Permission denied)?
 - If the same error is returned from every server, that's probably the one that should be returned, but need to define more error types and handle them in keepproxy.
- What is the correct response if the query gets a mix of 401, 403 and/or 404 errors?
 - Could happen if the site has a mix of servers with permissions enabled or disabled.

#6 - 10/19/2015 02:53 PM - Peter Amstutz

- In keepclient.go, define a new struct `KeepRequestError`
 - This should be equivalent to the `arvados.errors.KeepRequestError` in the Python SDK
 - Should contain a map of keep servers with the last error received resulting from contacting each keep server
- `getOrHead()` should return `BlockNotFound` if all servers replied http 404.
- `getOrHead()` should return `KeepRequestError` for any other error
- In keepproxy.go, `GetBlockHandler.ServeHTTP`, add a check for `KeepRequestError`. This should return a HTTP 422 status code and return a formatted text string of `KeepRequestError` in the body.

#7 - 10/19/2015 04:17 PM - Tom Clegg

- Description updated

#8 - 10/19/2015 04:38 PM - Tom Clegg

- Description updated

#9 - 10/19/2015 06:16 PM - Radhika Chippada

- Status changed from New to In Progress

#10 - 10/21/2015 01:57 PM - Radhika Chippada

All the details from the description are implemented as suggested. One exception:

"In all cases, return `err.Error()` in the response body"

Setting the body with error message is interfering with `"http.Error(resp, err.Error(), status)"` in the defer func (Panic: runtime error: invalid memory address or nil pointer dereference). This is resulting in errors in `KeepClient` which is expecting error and nil reader etc. So, I have not implemented this. Please clarify requirement around this.

#11 - 10/22/2015 08:36 PM - Peter Amstutz

Reviewing 7492-keepproxy-upstream-errors:

- Is there a reason why you are using

```
respErr.Error() == "Block not found"
```

instead of

```
respErr == keepclient.BlockNotFound
```

or

```
c.Check(err, Equals, keepclient.BlockNotFound)
```

?

- Instead of calling `kc.getSortedRoots(locator)` twice, suggest doing this in `getOrHead()`:

```
count404 := len(serversToTry)
```

```
...
if resp.StatusCode == 404 {
    count404--
}
...
if count404 == 0 {
}
```

- Can we add a test to keepproxy that checks for permanent and temporary errors from the perspective of the client talking to the proxy?

#12 - 10/23/2015 03:20 PM - Radhika Chippada

Is there a reason why you are using `respErr.Error() "Block not found"`

Updated to use `respErr keepclient.BlockNotFound` and `c.Check(err, Equals, keepclient.BlockNotFound)`. Much better now.

Instead of calling `kc.getSortedRoots(locator)` twice, suggest doing this in `getOrHead()` ...

Removed the unnecessary `getSortedRoots` call and used `numServers` variable instead. I did not want to make the change to `count404--` as you suggested. Doing so results in code such as `if count404 == 0 { set status code to 404 }`, which can be quite counter-intuitive even if I use a better variable name :)

Can we add a test to keepproxy that checks for permanent and temporary errors from the perspective of the client talking to the proxy?

Added one more test set to do Get with a bad token and expired token. If you have any other specific tests in mind, please let me know.

Thanks.

#13 - 10/23/2015 03:45 PM - Peter Amstutz

Radhika Chippada wrote:

Is there a reason why you are using `respErr.Error() "Block not found"`

Updated to use `respErr keepclient.BlockNotFound` and `c.Check(err, Equals, keepclient.BlockNotFound)`. Much better now.

Instead of calling `kc.getSortedRoots(locator)` twice, suggest doing this in `getOrHead()` ...

Removed the unnecessary `getSortedRoots` call and used `numServers` variable instead. I did not want to make the change to `count404--` as you suggested. Doing so results in code such as `if count404 == 0 { set status code to 404 }`, which can be quite counter-intuitive even if I use a better variable name :)

Thanks.

Can we add a test to keepproxy that checks for permanent and temporary errors from the perspective of the client talking to the proxy?

Added one more test set to do Get with a bad token and expired token. If you have any other specific tests in mind, please let me know.

Yes, can we have a test where keepproxy generates a `Temporary()` error? The original goal of this story was for keepproxy to start returning 502 Bad Gateway when one of the keep servers behind the proxy failed in a "temporary" way so as to trigger retry behavior on the external client.

#14 - 10/24/2015 08:02 PM - Radhika Chippada

Yes, can we have a test where keepproxy generates a `Temporary()` error ...

Added another test to `keepproxy_test` with "connection refused" temporary error. Also, improved test assertions in `keepclient_test`. Thanks.

#15 - 10/27/2015 05:20 PM - Peter Amstutz

Radhika Chippada wrote:

Yes, can we have a test where keepproxy generates a `Temporary()` error ...

Added another test to keepproxy_test with "connection refused" temporary error. Also, improved test assertions in keepclient_test. Thanks.

Unfortunately, that is not testing what I was looking to be tested. This is testing whether a failed connection to the proxy results in a Temporary error. I'm asking for a test where the client can connect to the proxy, but the proxy can't connect to the keep node.

#16 - 10/28/2015 04:17 PM - Peter Amstutz

[0bb6625](#) looks good to me.

#17 - 10/28/2015 04:23 PM - Radhika Chippada

- *Story points set to 1.0*

#18 - 10/28/2015 04:25 PM - Radhika Chippada

- *Status changed from In Progress to Resolved*

- *% Done changed from 0 to 100*

Applied in changeset arvados|commit:b90d3ea4986a1af59395f223e0320f07cc91c272.