

Arvados - Bug #13581

Report node not satisfiable users to the user

06/06/2018 03:14 PM - Tom Morris

Status:	Resolved	Start date:	06/07/2018
Priority:	Normal	Due date:	
Assigned To:	Lucas Di Pentima	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	2018-06-20 Sprint		
Description			
Similar to #7475 , container requests which are not satisfiable should be reported to the user in a useful way rather than just getting put in a system log for the sysadmin to dig out.			
Subtasks:			
Task # 13587: Review 13581-cr-not-satisfiable-report			Closed

Associated revisions

Revision 94f2b439 - 06/15/2018 07:59 PM - Lucas Di Pentima

Merge branch '13581-cr-not-satisfiable-report'
Closes #13581

Arvados-DCO-1.1-Signed-off-by: Lucas Di Pentima <ldipentima@veritasgenetics.com>

History

#1 - 06/06/2018 03:18 PM - Lucas Di Pentima

- Assigned To set to Lucas Di Pentima

#2 - 06/06/2018 07:13 PM - Lucas Di Pentima

- Status changed from New to In Progress

#3 - 06/07/2018 06:02 PM - Lucas Di Pentima

Updates at [652b52a64](#) - branch 13581-cr-not-satisfiable-report
Test run: <https://ci.curoverse.com/job/developer-run-tests/742/>

- Added return value to `dispatchcloud.ChooseInstanceType()` that states which resource wasn't satisfiable.
- Updated c-d-s to add this reason to the error message sent to the CR logs.

From the description of this bug report, it seems that the error message was only available from system logs, but the code to submit the user log was already there on c-d-s, are we sure that the error message wasn't available on the CR?

#4 - 06/11/2018 09:49 PM - Tom Clegg

The usual pattern for this is to customize the err returned by `ChooseInstanceType`, rather than requiring all of the intervening functions to manage two separate error values. Something like this:

```
type ConstraintsNotSatisfiableError struct {
    error
    AvailableTypes []arvados.InstanceType
}

// ...and instead of "if err == ErrConstraintsNotSatisfiable"...
if err, ok := err.(ConstraintsNotSatisfiableError); ok {
    for _, t := range err.AvailableTypes {
        log.Printf("Available instance type: %d RAM, ...", t.RAM, ...)
    }
}
```

I suspect the "max" calculation is wrong (`errReason` is just the reason the *last considered* node didn't qualify, so you might ask for 8 CPUs and get an error "max CPUs is 8" if the last type in the list had plenty of RAM but only 4 CPUs). I think this also means we need to think through how to report the "biggest" instance types.

For example, if the instance types are

- 512G RAM, 32 CPUs
- 256G RAM, 64 CPUs

...and a container wants 300G RAM + 40 CPUs, then it would be misleading to say either "unsatisfiable: max RAM is 256G" or "unsatisfiable: max CPUs is 32". To be helpful we'd really have to provide the specs for both of those "biggest" node types.

Rather than listing the maximum, we could offer "max RAM is 256G for an instance with 40 CPUs" and "max CPUs is 32 for an instance with 300G RAM" ...but this might get complicated with all 3 dimensions.

Maybe list all instance types that satisfy at least one of the given constraints, sorted by price? Or, if there are none, list all of the instance types?

When tracking max figures, use an `arvados.RuntimeConstraints` struct instead of a `map[string]int64`. Then you can say `max.Scratch < it.Scratch` instead of `max["Scratch"] < it.Scratch`.

#5 - 06/12/2018 01:54 PM - Tom Clegg

Lucas Di Pentima wrote:

From the description of this bug report, it seems that the error message was only available from system logs, but the code to submit the user log was already there on c-d-s, are we sure that the error message wasn't available on the CR?

Maybe try it on a dev cluster and see what happens...

#6 - 06/12/2018 05:28 PM - Lucas Di Pentima

Rebased & pushed updates at [6714a5e7fcf4d5fde2ecd5a7f9f6504cb5ca374b](https://github.com/Arvados/arvados/commit/6714a5e7fcf4d5fde2ecd5a7f9f6504cb5ca374b)

Following above suggestions here and from chat conversations, added all available instance types to the error returned by `ChooseInstanceType()`, and use that list to build the log message on the api server.

I'm not sure the way I'm checking for errors is the correct one at `file services/crunch-dispatch-slurm/crunch-dispatch-slurm_test.go` line 401, any pointers will be appreciated.

#7 - 06/12/2018 05:39 PM - Tom Morris

I should have included a reference to the case that triggered this request. It came from <https://dev.arvados.org/issues/13566#note-10>

Independent of the immediate issue, this error should be reported to the user in a way that they can understand and take action on. It looks like perhaps #7475 was intended to address this but when I look at https://workbench.su92l.arvadosapi.com/container_requests/su92l-xvhdp-15cz7wwz4unld2a#Log all I see is

```
2018-06-05T18:11:17.727941724Z arvados.cwl-runner INFO: [container workflow.json#main]
su92l-xvhdp-xixdb3bnscadi3q is Final
2018-06-05T18:11:17.749859224Z arvados.cwl-runner INFO: [container workflow.json#main] error log:
2018-06-05T18:11:17.750137524Z arvados.cwl-runner INFO: ** log is empty **
2018-06-05T18:11:17.750521724Z arvados.cwl-runner WARNING: Overall process status is permanentFail
2018-06-05T18:11:18.410633224Z arvados.cwl-runner INFO: Final output collection 282fbb8a925551343e98e762fa
1ec1be+57 "Output of main (2018-06-05T18:11:18.331Z)" (su92l-4zz18-k9iab1i5zmbvwmy)
2018-06-05T18:11:19.217568524Z cwltool WARNING: Final process status is permanentFail
```

#8 - 06/13/2018 12:46 PM - Lucas Di Pentima

Tom Morris wrote:

I should have included a reference to the case that triggered this request. It came from <https://dev.arvados.org/issues/13566#note-10>

Independent of the immediate issue, this error should be reported to the user in a way that they can understand and take action on. It looks like perhaps #7475 was intended to address this but when I look at https://workbench.su92l.arvadosapi.com/container_requests/su92l-xvhdp-15cz7wwz4unld2a#Log all I see is

[...]

Thanks Tom for the example. Checking its cancelled child container logs, it seems that the error was correctly reported:

https://workbench.su92l.arvadosapi.com/container_requests/su92l-xvhdp-xixdb3bnscadi3q#Log

This update will add to that log all the available instance types so the user can make a correction having the complete picture.

#9 - 06/14/2018 02:19 PM - Tom Clegg

Might be a bit nicer to use `FitsTypeOf` here in `lib/dispatchcloud/node_size_test.go`:

```
- err, ok := err.(ConstraintsNotSatisfiableError)
```

```
- c.Check(ok, check.Equals, true)
+ c.Check(err, check.FitsTypeOf, ConstraintsNotSatisfiableError{})
```

Nit: in services/crunch-dispatch-slurm/crunch-dispatch-slurm.go, logBuf.WriteString(fmt.Sprintf(...)) can also be written fmt.Fprintf(logBuf, ...)

In that same place, if len(err.AvailableTypes) == 0 then I think we would have a bug (we are supposed to just skip assigning an instance type if none are configured). So it might be better to *not* suppress the "Available instance types:" header so there's still a clue about what's happening -- or perhaps print a "no instance types are configured" or "(bug???)" message in its place.

#10 - 06/14/2018 03:06 PM - Lucas Di Pentima

Updates at [9f7687081](#)

Test run: <https://ci.curoverse.com/job/developer-run-tests/753/>

Added message when there aren't instance types configured. Also, updated test & buffer write code as suggested.

#11 - 06/15/2018 07:57 PM - Tom Clegg

LGTM, thanks!

#12 - 06/15/2018 08:37 PM - Lucas Di Pentima

- Status changed from *In Progress* to *Resolved*

Applied in changeset [arvados|94f2b439783a8e63d6d7b9ba2760f54fc642a8fb.](#)

#13 - 07/23/2018 06:52 PM - Tom Morris

- Release set to 13