FreeNAS - Bug #43811

Fix deadlock on thread pool

08/30/2018 11:28 AM - Waqar Ahmed

Status:	Done		
Priority:	No priority		
Assignee:	William Grzybowski		
Category:	Middleware		
Target version:	11.2-BETA3		
Seen in:	Master - FreeNAS Nightlies	**Needs Merging:**	No
Severity:	High	**Needs Automation:**	No
Reason for Closing:		**Support Suite Ticket:**	n/a
Reason for Blocked:		**Hardware Configuration:**	
Needs QA:	No	**ChangeLog Required:**	No
Needs Doc:	No		

Description

This issue has been seen on a machine in TN Office (realmini), it seems randomly middlewared becomes totally unresponsive and locks itself out. The issue is resolved for the time being by restarting middlewared but the root cause should be investigated and patched.

Risk
This is already a high risk ticket and is causing total failure of middlewared.

Acceptance Criteria
Running
"seq 1 25|while read i; do midclt call vm.get_available_memory &; done;"

should finish as expected and not lock up middlewared.

Related issues:

Related to FreeNAS - Bug #43789: Sentry responded with an API error: RateLimi...	**Closed**

Associated revisions

Revision c92650c2 - 08/30/2018 12:27 PM - William Grzybowski

fix(middlewared): deadlock on thread pool

Threads to handle websocket connection are gated on `__threadpool`.
Any other calls should use `run_in_io_thread` as that launches its own thread
and does not cause deadlock waiting another thread to finish in the pool
(which could happen on the stack call, e.g.
service.foo calls something in using the thread pool and something also
uses the thread pool. If service.foo is called many times before each thread
finishes we will have a deadlock)

Ticket: #43811

Revision 3d9da0ea - 08/30/2018 12:29 PM - William Grzybowski

fix(middlewared): deadlock on thread pool

Threads to handle websocket connection are gated on `__threadpool`.
Any other calls should use `run_in_io_thread` as that launches its own thread
and does not cause deadlock waiting another thread to finish in the pool
(which could happen on the stack call, e.g.
service.foo calls something in using the thread pool and something also
uses the thread pool. If service.foo is called many times before each thread
finishes we will have a deadlock)

Ticket: #43811

Revision 39fcd1d3 - 08/30/2018 12:40 PM - William Grzybowski

fix(middlewared): deadlock on thread pool

Threads to handle websocket connection are gated on `__threadpool`.
Any other calls should use `run_in_io_thread` as that launches its own thread
and does not cause deadlock waiting another thread to finish in the pool
(which could happen on the stack call, e.g.
service.foo calls something in using the thread pool and something also
uses the thread pool. If service.foo is called many times before each thread
finishes we will have a deadlock)

Ticket: #43811

Revision 1a2664f0 - 08/30/2018 12:42 PM - William Grzybowski

fix(middlewared): deadlock on thread pool

Threads to handle websocket connection are gated on `__threadpool`.
Any other calls should use `run_in_io_thread` as that launches its own thread
and does not cause deadlock waiting another thread to finish in the pool
(which could happen on the stack call, e.g.
service.foo calls something in using the thread pool and something also
uses the thread pool. If service.foo is called many times before each thread
finishes we will have a deadlock)

Ticket: #43811

Revision 2358dac5 - 08/30/2018 01:06 PM - William Grzybowski

fix(middlewared): deadlock on thread pool

Threads to handle websocket connection are gated on `__threadpool`.
Any other calls should use `run_in_io_thread` as that launches its own thread
and does not cause deadlock waiting another thread to finish in the pool
(which could happen on the stack call, e.g.
service.foo calls something in using the thread pool and something also
uses the thread pool. If service.foo is called many times before each thread
finishes we will have a deadlock)

Ticket: #43811

Revision 1f3450c6 - 08/30/2018 01:08 PM - William Grzybowski

fix(middlewared): deadlock on thread pool

Threads to handle websocket connection are gated on `__threadpool`.
Any other calls should use `run_in_thread` as that launches its own thread
and does not cause deadlock waiting another thread to finish in the pool
(which could happen on the stack call, e.g.
service.foo calls something in using the thread pool and something also
uses the thread pool. If service.foo is called many times before each thread
finishes we will have a deadlock)

Ticket: #43811

Revision 274e4555 - 08/30/2018 01:13 PM - William Grzybowski

fix(middlewared): deadlock on thread pool

Threads to handle websocket connection are gated on `__threadpool`.
Any other calls should use `run_in_thread` as that launches its own thread
and does not cause deadlock waiting another thread to finish in the pool
(which could happen on the stack call, e.g.
service.foo calls something in using the thread pool and something also
uses the thread pool. If service.foo is called many times before each thread
finishes we will have a deadlock)

Ticket: #43811

Revision f9f38d0b - 08/30/2018 01:18 PM - William Grzybowski

fix(middlewared): deadlock on thread pool

Threads to handle websocket connection are gated on `__threadpool`.
Any other calls should use `run_in_thread` as that launches its own thread
and does not cause deadlock waiting another thread to finish in the pool
(which could happen on the stack call, e.g.

service.foo calls something in using the thread pool and something also uses the thread pool. If service.foo is called many times before each thread finishes we will have a deadlock)

Ticket: #43811

Revision 4a5d8a07 - 09/03/2018 08:25 AM - William Grzybowski

fix(middlewared/disk): boot.get_disks is no longer an async generator

Ticket: #43811

Revision 9fdb26e4 - 09/03/2018 08:29 AM - William Grzybowski

fix(middlewared/disk): boot.get_disks is no longer an async generator

Ticket: #43811

History

#1 - 08/30/2018 11:30 AM - William Grzybowski

- Status changed from Unscreened to Not Started

- Severity changed from New to High

#2 - 08/30/2018 11:33 AM - Dru Lavigne

- Target version changed from Backlog to 11.2-BETA3

#3 - 08/30/2018 12:31 PM - Bug Clerk

- Status changed from Not Started to In Progress

#4 - 08/30/2018 12:32 PM - William Grzybowski

- Description updated

#5 - 08/30/2018 12:59 PM - William Grzybowski

- Related to Bug #43789: Sentry responded with an API error: RateLimited(None) added

#6 - 08/30/2018 01:19 PM - Bug Clerk

- Status changed from In Progress to Ready for Testing

PR: https://github.com/freenas/freenas/pull/1738

#7 - 08/31/2018 05:36 AM - Dru Lavigne

- Subject changed from Middlewared becomes totally unresponsive to Fix deadlock on thread pool

- Needs Doc changed from Yes to No

- Needs Merging changed from Yes to No

#8 - 08/31/2018 07:11 AM - Bonnie Follweiler

- Status changed from Ready for Testing to Blocked

- Reason for Blocked set to Waiting for feedback

#11 - 08/31/2018 10:01 AM - Bonnie Follweiler

- Status changed from Blocked to Passed Testing

- Reason for Blocked deleted (Waiting for feedback)

- Needs QA changed from Yes to No

Test passed in FreeNAS-11.2-MASTER-201808310859

#12 - 08/31/2018 02:40 PM - Dru Lavigne

- Status changed from Passed Testing to Done

#13 - 09/03/2018 08:26 AM - Bug Clerk

- Status changed from Done to In Progress

#14 - 09/03/2018 08:29 AM - Bug Clerk

- Status changed from In Progress to Ready for Testing

PR: https://github.com/freenas/freenas/pull/1746

#15 - 09/03/2018 08:30 AM - William Grzybowski

- Status changed from Ready for Testing to Done