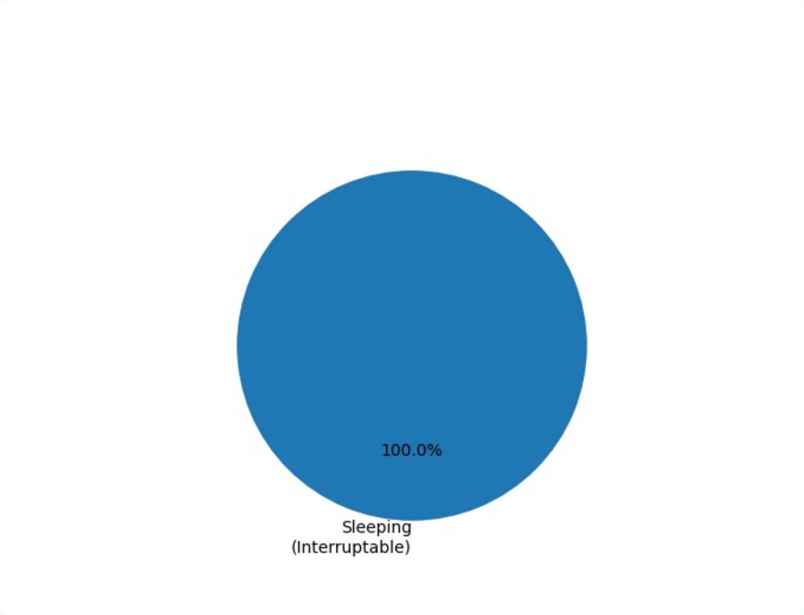
Stack Trace report for eu-stack of the mongod server

Thread State Distribution Graph

Pie graph to illustrate the different states of threads present



Thread State	Thread Count
Sleeping (Interruptable)	2

Thread Details

Shows details of each thread

Thread ID	Thread Name	Thread State	Thread CPU
6882	conn1	S	0.0
6888	conn2	S	0.0

Flame Graph

Shows the call stack in the form of a tree

Identical Stack Traces Distribution

Shows statistics related to frequency of stack traces amongst different threads. Refer to below table for actual stack values

Identical Stack Traces Distribution

Stack Name	Stack Trace	Thread Count	Thread List
S1	#0 0x00007fc02f3fd9ef recvmsg #1 0x000055d94d61c522 asio::detail::socket_ops::recv(int, iov....	2	6882, 6888

Most Used Functions

Shows which functions have been most used across the entire stack trace

Thread		

Count	Function Name	Percentage
2	recvmsg	5.9
2	asio::detail::socket_ops::recv(int, iovec*, unsigned long, int, std::error_code&)	5.9
2	asio::detail::socket_ops::sync_recv(int, unsigned char, iovec*, unsigned long, int, bool, std::error_code&)	5.9
2	mongo::Future mongo::transport::TransportLayerASIO::ASIOSession::opportunisticRead, asio::mutable_buffers_1> (asio::basic_stream_socket&, asio::mutable_buffers_1 const&, std::shared_ptr const&)	5.9
2	mongo::Future mongo::transport::TransportLayerASIO::ASIOSession::read(asio::mutable_buffers_1 const&, std::shared_ptr const&)	5.9
2	mongo::transport::TransportLayerASIO::ASIOSession::sourceMessageImpl(std::shared_ptr const&)	5.9
2	mongo::transport::TransportLayerASIO::ASIOSession::sourceMessage()	5.9
2	mongo::transport::ServiceStateMachine::Impl::sourceMessage()	5.9
2	mongo::transport::ServiceStateMachine::Impl::startNewLoop(mongo::Status const&)	5.9
2	mongo::unique_function::makeImpl(mongo::transport::ServiceStateMachine::Impl::scheduleNewLoop(mongo::Status):: {lambda(mongo::Status)#1}&&)::SpecificImpl::call(mongo::Status&&)	5.9
2	mongo::unique_function::makeImpl const&, mongo::unique_function)::{lambda(mongo::Status)#1}> (mongo::transport::ServiceExecutorSynchronous::runOnDataAvailable(std::shared_ptr const&, mongo::unique_function):: {lambda(mongo::Status)#1}&&)::SpecificImpl::call(mongo::Status&&)	5.9
2	mongo::unique_function::makeImpl)::{lambda()#1}>(mongo::transport::ServiceExecutor::schedule(mongo::unique_function):: {lambda()#1}&&)::SpecificImpl::call()	5.9
2	mongo::unique_function::makeImpl, mongo::transport::ServiceExecutor::ScheduleFlags)::{lambda()#2}> (mongo::transport::ServiceExecutorSynchronous::scheduleTask(mongo::unique_function, mongo::transport::ServiceExecutor::ScheduleFlags)::{lambda()#2}&&)::SpecificImpl::call()	5.9
2	mongo::unique_function::makeImpl)::{lambda()#4}>(mongo::launchServiceWorkerThread(mongo::unique_function):: {lambda()#4}&&)::SpecificImpl::call()	5.9
2	mongo::(anonymous namespace)::runFunc(void*)	5.9
2	start_thread	5.9
2	__clone3	5.9

Top CPU Consuming Threads (Top 5)

Thread ID	Thread Name	Thread State	Thread CPU	Thread Stack
6882	conn1	S	0.0	#0 0x00007fc02f3fd9ef recvmsg #1 0x000055d94d61c522 asio::detail::socket_ops::recv(int, iovec*, unsigned long, int, std::error_code&) #2 0x000055d94d61c5d8 asio::detail::socket_ops::sync_recv(int, unsigned char, iovec*, unsigned long, int, bool, std::error_code&) #3 0x000055d94d15b073 mongo::Future mongo::transport::TransportLayerASIO::ASIOSession::opportunisticRead, asio::mutable_buffers_1>(asio::basic_stream_socket&, asio::mutable_buffers_1 const&, std::shared_ptr const&) #4 0x000055d94d1691aa mongo::Future mongo::transport::TransportLayerASIO::ASIOSession::read(asio::mutable_buffers_1 const&, std::shared_ptr const&) #5 0x000055d94d1526b7 mongo::transport::TransportLayerASIO::ASIOSession::sourceMessageImpl(std::shared_ptr const&) #6 0x000055d94d152a0a mongo::transport::TransportLayerASIO::ASIOSession::sourceMessage() #7 0x000055d94b08500d mongo::transport::ServiceStateMachine::Impl::sourceMessage() #8 0x000055d94b086e72 mongo::transport::ServiceStateMachine::Impl::startNewLoop(mongo::Status const&) #9 0x000055d94b08722f mongo::unique_function::makeImpl(mongo::transport::ServiceStateMachine::Impl::scheduleNewLoop(mongo::Status):: {lambda(mongo::Status)#1}&&)::SpecificImpl::call(mongo::Status&&) #10 0x000055d94d1229c1 mongo::unique_function::makeImpl const&, mongo::unique_function):: {lambda(mongo::Status)#1}>(mongo::transport::ServiceExecutorSynchronous::runOnDataAvailable(std::shared_ptr const&, mongo::unique_function)::{lambda(mongo::Status)#1}&&)::SpecificImpl::call(mongo::Status&&) #11 0x000055d94b766ed3 mongo::unique_function::makeImpl)::{lambda()#1}> (mongo::transport::ServiceExecutor::schedule(mongo::unique_function)::{lambda()#1}&&)::SpecificImpl::call() #12 0x000055d94d122b7b mongo::unique_function::makeImpl, mongo::transport::ServiceExecutor::ScheduleFlags):: {lambda()#2}>(mongo::transport::ServiceExecutorSynchronous::scheduleTask(mongo::unique_function, mongo::transport::ServiceExecutor::ScheduleFlags)::{lambda()#2}&&)::SpecificImpl::call() #13 0x000055d94d126a2c mongo::unique_function::makeImpl)::{lambda()#4}> (mongo::launchServiceWorkerThread(mongo::unique_function)::{lambda()#4}&&)::SpecificImpl::call() #14 0x000055d94d126a9c mongo::(anonymous namespace)::runFunc(void*) #15 0x00007fc02f36ab43 start_thread #16 0x00007fc02f3fca00 __clone3
				#0 0x00007fc02f3fd9ef recvmsg #1 0x000055d94d61c522 asio::detail::socket_ops::recv(int, iovec*, unsigned long, int, std::error_code&) #2 0x000055d94d61c5d8 asio::detail::socket_ops::sync_recv(int, unsigned char, iovec*, unsigned long, int, bool, std::error_code&) #3 0x000055d94d15b073 mongo::Future mongo::transport::TransportLayerASIO::ASIOSession::opportunisticRead, asio::mutable_buffers_1>(asio::basic_stream_socket&, asio::mutable_buffers_1 const&, std::shared_ptr const&) #4 0x000055d94d1691aa mongo::Future mongo::transport::TransportLayerASIO::ASIOSession::read(asio::mutable_buffers_1 const&, std::shared_ptr const&) #5 0x000055d94d1526b7 mongo::transport::TransportLayerASIO::ASIOSession::sourceMessageImpl(std::shared_ptr const&)

6888	conn2	S	0.0	#6 0x000055d94d152a0a mongo::transport::TransportLayerASIO::ASIOSession::sourceMessage() #7 0x000055d94b08500d mongo::transport::ServiceStateMachine::Impl::sourceMessage() #8 0x000055d94b086e72 mongo::transport::ServiceStateMachine::Impl::startNewLoop(mongo::Status const&) #9 0x000055d94b08722f mongo::unique_function::makeImpl(mongo::transport::ServiceStateMachine::Impl::scheduleNewLoop(mongo::Status)::{lambda(mongo::Status)#1}&&)::SpecificImpl::call(mongo::Status&&) #10 0x000055d94d1229c1 mongo::unique_function::makeImpl const&, mongo::unique_function)::{lambda(mongo::Status)#1}>(mongo::transport::ServiceExecutorSynchronous::runOnDataAvailable(std::shared_ptr const&, mongo::unique_function)::{lambda(mongo::Status)#1}&&)::SpecificImpl::call(mongo::Status&&) #11 0x000055d94b766ed3 mongo::unique_function::makeImpl)::{lambda()#1}>(mongo::transport::ServiceExecutor::schedule(mongo::unique_function)::{lambda()#1}&&)::SpecificImpl::call() #12 0x000055d94d122b7b mongo::unique_function::makeImpl, mongo::transport::ServiceExecutor::ScheduleFlags)::{lambda()#2}>(mongo::transport::ServiceExecutorSynchronous::scheduleTask(mongo::unique_function, mongo::transport::ServiceExecutor::ScheduleFlags)::{lambda()#2}&&)::SpecificImpl::call() #13 0x000055d94d126a2c mongo::unique_function::makeImpl)::{lambda()#4}>(mongo::launchServiceWorkerThread(mongo::unique_function)::{lambda()#4}&&)::SpecificImpl::call() #14 0x000055d94d126a9c mongo::(anonymous namespace)::runFunc(void*) #15 0x00007fc02f36ab43 start_thread #16 0x00007fc02f3fca00 __clone3