RSBag - Bug #1170

Recording video with rsbag-tools-cl 0.7 with rst injected crashes shortly after recording start

09/20/2012 11:21 AM - J. Wienke

Status:	Closed	**Start date:**	09/20/2012
Priority:	Normal	**Due date:**	
Assignee:	J. Moringen	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	rsb-0.7		

Description

languitar@herbie:~/Desktop/bag-tools/0.7$ ./bag-record -o vision0.tide spread:/nao/vision/0?name=4803
2012-09-20T11:19:40.838863+02:00 #<SYNCHRONIZED-CHANNEL "/nao/vision/0/:.rst.vision.Image" (93) (RSB-EVENT-0.7 .rst.vision.Image) {100DA4D903}>
[WARN (RSB.COMMON)] Aborting background thread #<THREAD "Message Receiver for #<IN-PUSH spread::4803/ {100D1BFA23}>" RUNNING {100B0C8CF3}>
NIL doesn't designate a condition class.
[WARN (RSBAG.TOOLS.RECORD)] Error during detaching of #<LISTENER /nao/vision/0/ |(0) D8DEDED5>: Interrupt thread failed: thread #<THREAD "Message Receiver for #<IN-PUSH spread::4803/ {100D1BFA23}>" ABORTED {100B0C8CF3}> has exited.
languitar@herbie:~/Desktop/bag-tools/0.7$ ./bag-info vision0.tide
File "vision0.tide"
  Events : 165
  Start   : 2012-09-20T11:19:30.343093+02:00
  End   : 2012-09-20T11:19:39.713075+02:00
  Duration: 9.369982
  Channel "/nao/vision/0/:.rst.vision.Image"
    Type   : (RSB-EVENT-0.7 .rst.vision.Image)
    Format : // Notification.proto
    package rsb.protocol;
    import "rsb/protocol/EventId.proto";
    import "rsb/protoco…
    Events : 165
    Start   : 2012-09-20T11:19:30.343093+02:00
    End   : 2012-09-20T11:19:39.713075+02:00
    Duration: 9.369982
    Rate   : 17.60942550369894

languitar@herbie:~/Desktop/bag-tools/0.7$ ./bag-info --version
./bag-info version   0.7.0
SBCL version   1.0.57
RSB version   0.7.0
RSBAG version   0.7.0
RSBAG-TIDELOG version 0.7.0

Subtasks:

Bug # 1172: Condition propragation between threads fails	**Resolved**

Associated revisions

Revision 974e19ba - 09/30/2012 10:11 AM - J. Moringen

Improved error-handling functions in common/error-handling.lisp

refs #1170, fixes #1172

- common/error-handling.lisp (abort/signal): made condition parameter
  mandatory; call `invoke-restart' with the restart object
  (continue/verbose): fixed logging format strings
  (maybe-relay-to-thread): do not provide an `abort' restart in the
  target thread; do not call the policy in the target thread if it
  would abort anyway; provide `abort/signal' restart in worker threads;
  renamed parameter strategy -> policy
  (ftype invoke-with-error-policy): fixed function name
  invoke-with-error-handling-strategy -> invoke-with-error-policy
  (invoke-with-error-policy): use `restart-case' instead of
  `restart-bind' to ensure resignaling outside the dynamic scope of
  the error policy which may otherwise apply relay or logging behavior
  a second time

Revision a50a011b - 01/21/2013 02:48 PM - J. Moringen

Fixed performance regression in src/transport/spread/fragmentation.lisp

refs #1170

A type declaration using `octet-vector' used to be correct. However,
in nibbles, there are `nibbles:octet-vector' and
`nibbles:simple-octet-vector'. The declaration should have been
changed to the latter.

- src/transport/spread/fragmentation.lisp
  (assembly-concatenated-data): specify "fast and unsafe" optimization
  policy; fixed fragment type declaration octet-vector ->
  simple-octet-vector

Revision 6cdef583 - 01/21/2013 02:58 PM - J. Moringen

Fixed performance regression in src/transport/spread/fragmentation.lisp

refs #1170

A type declaration using `octet-vector' used to be correct. However,
in nibbles, there are `nibbles:octet-vector' and
`nibbles:simple-octet-vector'. The declaration should have been
changed to the latter.

- src/transport/spread/fragmentation.lisp
  (assembly-concatenated-data): specify "fast and unsafe" optimization
  policy; fixed fragment type declaration octet-vector ->
  simple-octet-vector

Revision acfd9a5d - 01/21/2013 03:16 PM - J. Moringen

Backport: Fixed performance regression in src/transport/spread/fragmentation.lisp

refs #1170

A type declaration using `octet-vector' used to be correct. However,
in nibbles, there are `nibbles:octet-vector' and
`nibbles:simple-octet-vector'. The declaration should have been
changed to the latter.

- src/transport/spread/fragmentation.lisp
  (assembly-concatenated-data): specify "fast and unsafe" optimization
  policy; fixed fragment type declaration octet-vector ->
  simple-octet-vector

Revision d2f30c9e - 02/13/2013 05:11 PM - J. Moringen

Backport: Fixed performance regression in src/transport/spread/fragmentation.lisp

refs #1170

A type declaration using `octet-vector' used to be correct. However,
in nibbles, there are `nibbles:octet-vector' and
`nibbles:simple-octet-vector'. The declaration should have been
changed to the latter.

- src/transport/spread/fragmentation.lisp
  (assembly-concatenated-data): specify "fast and unsafe" optimization
  policy; fixed fragment type declaration octet-vector ->
  simple-octet-vector

Revision a1f60216 - 02/15/2013 08:22 PM - J. Moringen

Load network.spread instead of cl-spread in main/dump.lisp

refs #1170

- main/dump.lisp: load system network.spread instead of cl-spread; use
  `network.spread:enable-reload-spread-library' instead of
  unloading/reloading the Spread library directly

Revision 2d974c91 - 02/15/2013 08:23 PM - J. Moringen

Use rsbag:enable-restart-threadpool in main/dump.lisp

refs #1170, refs #1013

- main/dump.lisp: when saving and resuming an image, the rsbag
  threadpool has to be stopped and restarted; call
  `rsbag:enable-restart-threadpool' to achieve this

Revision add046ff - 02/15/2013 10:51 PM - J. Moringen

Decrease memory pressure in src/backend/tidelog/file.lisp

refs #852, refs #1170

- src/backend/tidelog/file.lisp (header): added one-line summary;
  updated copyright
  (write-buffer file chnk): after the buffer has been written
  overwrite references to chunk entries to allow earlier garbage
  collection; on SBCL, do garbage collection afterwards

Revision 1e27cd43 - 02/15/2013 10:53 PM - J. Moringen

Improved writing performance of TIDELog backend

refs #1170

- `pack' methods accept a stream as destination, avoiding the need for
  a temporary buffer for serialization
- The new generic function `tag' returns a statically allocated
  octet-vector containing the "tag" for a given block class (or
  object), avoiding going through symbol-based computations to produce
  the tag at runtime

- src/backend/tidelog/io.lisp (header): updated copyright
  (pack standard-object stream): removed; no longer needed
  (pack cons stream): likewise
- src/backend/tidelog/generator.lisp (header): updated copyright
  (specs->class): accept new keyword parameter toplevel?; if supplied,
  emit methods on `tag' which return the tag for class being defined
  (type-spec->lisp-type): cosmetic changes
  (specs->size): likewise
  (spec->size): likewise
  (specs->deserializer): likewise
  (specs->serializer): emit two methods instead of one; one for array
  destinations and one for stream destinations; accept new toplevel?
  keyword parameter; if non-nil generate an :around method on `pack'
  which writes a block header
  (spec->serializer): accept a medium parameter which controls whether
  to generate for array destinations or stream destinations
  (type-spec->serializer/buffer): renamed type-spec->serializer ~~->~~
  ~~type-spec~~>serializer/buffer
  (type-spec->serializer/stream): like `type-spec->serializer/buffer',
  but for stream destination
- src/backend/tidelog/macros.lisp (header): updated copyright
  (define-element): accept :toplevel? option, pass to `specs->class'
  and `specs->serializer'
- src/backend/tidelog/spec.lisp (header): updated copyright
  (define-element tide): added :toplevel? option
  (define-element chan): likewise
  (define-element indx): likewise

(define-element chnk): likewise

Revision 316282c6 - 02/17/2013 06:10 PM - J. Moringen

Added async buffer write-backs in src/backend/backend-mixins.lisp

refs #1170, fixes #1013

- src/threadpool.lisp: new file; contains functions for managing a dedicated threadpool for rsbag
- src/reloading.lisp: new file; contains functions for stopping and restarting the rsbag threadpool when saving and loading an image
- src/package.lisp (header): updated copyright
  (package rsbag): added exported symbols start-threadpool, stop-threadpool, enable-restart-threadpool and with-threadpool
- src/backend/backend-mixins.lisp (header): updated copyright
  (**async?**): new variable; allows or disallows async write-back
  (async-double-buffered-writer-mixin): new class; adds async write-back behavior to buffered backend classes
  (shared-initialize :after async-double-buffered-writer-mixin): new method; allocate back-buffer
  (close :around async-double-buffered-writer-mixin): new method; force async operations to finish
  (write-buffer :around async-double-buffered-writer-mixin t): new method; initiate async write-back
  (make-buffer :around async-double-buffered-writer-mixin t): new method; create additional buffer
- src/backend/package (header): updated copyright
  (package rsbag.backend): added exported symbol async-double-buffered-writer-mixin
- src/backend/tidelog/file.lisp (file): added superclass `async-double-buffered-writer-mixin'
- test/backend/mixins.lisp: new file; contains unit tests for backend mixin classes
- test/backend/package.lisp (header): updated copyright
  (package rsbag.backed.test): added used package let-plus
- cl-rsbag.asd (header): updated copyright
  (system cl-rsbag): added system dependency on system lparallel; added files src/threadpool.lisp and src/reloading.lisp
  (system cl-rsbag-test): added file test/backend/mixins.lisp

Revision f33e1f02 - 02/17/2013 07:23 PM - J. Moringen

Changed system dependency cl-spread -> network.spread

refs #1170

- src/transport/spread/connection.lisp (header): updated copyright
  (connection::connection): changed type spread:connection -> network.spread:connection

(shared-initialize :after connection): changed package qualification spread -> network.spread
(ref-group connection string): likewise
(unref-group connection string): likewise
(receive-message connection t): likewise
(send-message connection list simple-array): likewise

- src/transport/spread/connector.lisp (header): updated copyright
(option connection::port): changed package qualification spread -> network.spread
(shared-initialize :after connector t): likewise
- test/transport/spread/connection.lisp (header): updated copyright
(spread-connection-root::construct): changed package qualification spread -> network.spread
- cl-rsb.asd (header): updated copyright
(system cl-rsb-doc): changed system dependency cl-spread -> network.spread
(system cl-rsb-test): likewise
(perform test-op eql find-system :cl-rsb-test): changed package qualification spread -> network.spread
(system cl-rsb-and-network.spread): renamed cl-rsb-and-cl-spread -> cl-rsb-and-network.spread; changed system dependency cl-spread -> network.spread

Revision a014ee82 - 02/17/2013 08:08 PM - J. Moringen

Receive into persistent buffer in src/transport/spread/connection.lisp

refs #1170

This reduces memory pressure and eliminates one copy operation.

- src/transport/spread/fragmentation.lisp (make-data-fragment): input and output buffers are `simple-octet-vector' instead of `octet-vector'
(%make-key): produce a `simple-octet-vector' instead of an `octet-vector'
- src/transport/spread/connection.lisp (connection::receive-buffer): new slot; stores a persistent, lazily allocated buffer which is used to receive Spread messages
(receive-message connection t): receive into persistent buffer
(send-message connection list simple-array): change type check octet-vector -> simple-octet-vector
(%ensure-receive-buffer): new function; helper function for lazily allocating receive buffer
- src/transport/spread/in-connector.lisp (header): updated copyright
(receive-message in-connector t): receive buffer and length as multiple values; return as cons
(message->event in-connector cons t): changed specializer simple-array -> cons; pass buffer and length to `pb:unpack'
- test/transport/spread/connectors.lisp (suite in-connector-root): adjusted mock messages to changed protocol
(test in-connector-root::message->event): simplified

- test/transport/spread/package.lisp
  (package rsb.transport.spread.test): added used package let-plus

Revision 9893aae7 - 02/17/2013 09:11 PM - J. Moringen

Load network.spread instead of cl-spread in main/dump.lisp

refs #1170

- main/dump.lisp: load system network.spread instead of cl-spread; use
  `network.spread:enable-reload-spread-library' instead of
  unloading/reloading the Spread library directly

History

#1 - 09/20/2012 02:52 PM - J. Moringen

- Subject changed from Recording video with rsbag-tools-cl 0.7 with rst injected crashes shortly afterrecording start to Recording video with rsbag-tools-cl 0.7 with rst injected crashes shortly after recording start

I could reproduce the problem (without the -rst-injector version). This is just an instance of the usual Spread disconnect problem.

The error being reported in this way is another issue (see subtask, #1172).

In my test, the disconnect occurred when writing a chunk to disk. Tweaking the buffering options may help. Merging the async buffer write-back patch will probably mitigate this problem to some extend.

#2 - 09/20/2012 03:22 PM - J. Moringen

At least in the scenario, I used to reproduce this, the problem seems to be related to Spread:

- Even if the informer slows down (to 50 % of the original speed) such that all processes stay below 15 % CPU and writes to /dev/null, the Spread connection is still terminated after a few seconds
- On the other hand, the socket transport, when writing to /dev/null, can handle the informer at 800 % of the original speed

#3 - 09/20/2012 04:11 PM - J. Wienke

As for HUMAVIPS we are now on 0.7 and socket seems to be still a problem (#1173), I would like to get this working stable in 0.7. Can you merge the patches back to 0.7?

#4 - 09/22/2012 07:48 AM - J. Moringen

> *[...] I would like to get this working stable in 0.7. Can you merge the patches back to 0.7?*

I think you misunderstood. The patch causes

    NIL doesn't designate a condition class.

to be replaced with the correct error message. It does not solve the Spread problem.

I will probably still push it later today.

#5 - 09/22/2012 08:24 AM - J. Wienke

ok, but in any case this used to be much more reliable in older versions. we could not even record a single video channel. we need the old reliability back.

#6 - 09/22/2012 08:30 AM - J. Moringen

This reminds me of the "bagder timeout" (or whatever it was called) problem in Spread.

If you think, this is a regression in cl-{rsb,rsbag}{,-tools}, we can try to bisect in two weeks.

#7 - 09/22/2012 08:48 AM - J. Moringen

At least on my system, the 0.6 version of bag-record has the same Spread problem (but reports the error correctly):

```
$ ./bag-record --version
./bag-record version  0.6.0
SBCL version          1.0.57
RSB version           0.6.0
RSBAG version         0.6.0
RSBAG-TIDELOG version 0.6.0
$ ./bag-record -o vision.tide spread:?name=4803
[~ 10 seconds later]
Error receiving: CONNECTION-CLOSED
2012-09-22T08:46:04.725496+02:00 #<SYNCHRONIZED-CHANNEL "/nao/vision/top/:.rst.vision.Image" (93) (RSB-EVENT-0.6
.rst.vision.Image) {109CB311}>
^C[ WARN (RSBAG.TOOLS.RECORD) ]  Error during detaching of #<LISTENER / |(0) 450B7F75>: Error leaving group
"6666cd76f96956469e7be39d750cc7d": NET-ERROR-ON-SESSION
$
```

#8 - 09/24/2012 10:11 AM - J. Wienke

Jan Moringen wrote:

> *This reminds me of the "bagder timeout" (or whatever it was called) problem in Spread.*

This could be true. I think we completely forgot about this issue. I will check this as soon as possible.

#9 - 02/08/2013 01:26 PM - J. Moringen

This is most likely caused by the badger-timeout problem. However, project:rsbag and in particular bag-record will get a few performance

improvements anyway.

#10 - 02/18/2013 04:35 PM - J. Moringen

- Status changed from New to Closed

0.7 and master:
Adjusting the badger timeout by patching the Spread daemon solves the instance of this problem in which bag-record cannot keep up despite appearing almost idle.

only master:
The performance improvements in the associated commits should mitigate most instances of this problem in which bag-record could genuinely not keep up.

Since there is already lots of only partially coherent stuff in this issue, I suggest opening a new issue if a similar problem appears again.