RSBag - Enhancement #846
Flush index blocks?

02/02/2012 03:52 PM - J. Moringen

Status:	Resolved	**Start date:**	02/02/2012
Priority:	Normal	**Due date:**	
Assignee:	J. Moringen	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	rsb-0.7		

Description

When recording a large number of events (i.e. millions), in-memory index blocks get rather big. It seems possible to regularly write back indices since fragmented indices have to be assembled when reading TIDELog anyway (or rather: this appears to be the case).

This would eliminate the biggest cause of unbounded memory consumption in the recording process.

Related issues:

Related to RSBag - Tasks # 847: Make buffer write back configurable	**Resolved**	**02/02/2012**
Related to RSBag - Enhancement # 1038: Allow composite flush strategies	**Resolved**	**06/26/2012**

Associated revisions

Revision f2a41dc4 - 06/20/2012 08:09 PM - J. Moringen

Added flush strategies in src/backend/flush-strategies.lisp

refs #847, #846

- src/protocol.lisp (open-bag): added flush-strategy keyword parameter
  (open-bag stream): accept flush-strategy keyword argument; pass to backend unless direction is input
- src/backend/protocol.lisp (buffer-property): new generic function; return specified property of a buffer
  (flush): new generic function; flush a buffer to background storage
  (backend-flush-strategy): new generic function; return the flush strategy associated with a backend
  (flush?) new generic function; return non-nil when a buffer should be flushed
  (define-findable-class-family flush-strategy): new findable class family; consists of flush strategy classes
  (make-flush-strategy): new generic function; make and return a flush strategy according to a specification
  (make-flush-strategy symbol): new method; construct from keyword or class name
  (make-flush-strategy class): new method; construct from given class
- src/backend/buffering-writer-mixin.lisp
  (buffering-writer-mixin::flush?-func): removed slot
  (buffering-writer-mixin::flush-strategy): new slot; stores flush strategy
  (close): call `flush' instead of `write-buffer'
  (put-entry buffering-writer-mixin t t t): call `flush?' and `flush'
  (flush buffering-writer-mixin t): new method; dispatch to `write-buffer'
  (flush :after buffering-writer-mixin t): changed write-buffer -> flush
- src/backend/flush-strategies.lisp: new file; contains flush strategies

- src/backend/package.lisp (package rsbag.backend): added used package more-conditions; added exported symbols buffer-property, flush, backend-flush-strategy, flush?, no-such-flush-strategy-class, find-flush-strategy-class, flush-strategy-classes and make-flush-strategy
- src/backend/tidelog/file.lisp (file): change default initargs to supply a flush strategy instead of a flush? function (buffer-property file chnk eql :length/entries): new method; return number of entries in chunk (buffer-property file chnk eql :length/bytes): similar for bytes (buffer-size->): removed; no longer required
- test/protocol.lisp (test suite protocol-root): added local functions pathname/existing, namestring/existing, stream (test protocol-root::open-bag/valid): use these; test flush strategies (test protocol-root::open-bag/invalid): likewise
- test/backend/flush-strategies.lisp: new file; contains tests for flush strategies
- cl-rsbag.asd (system cl-rsbag): added file src/backend/flush-strategies.lisp (system cl-rsbag-test): added file test/backend/flush-strategies.lisp

Revision 7e8ef83c - 06/21/2012 02:28 AM - J. Moringen

Use buffering-writer-mixin in src/backend/tidelog/index.lisp

refs #846

- src/backend/tidelog/index.lisp (header): added one-line summary (timestamps): use `missing-required-initarg' instead of `required-initarg' (index): added superclass `buffering-writer-mixin'; use `missing-required-initarg' instead of `required-initarg'; add default initarg for flush-strategy (index::buffer): removed; provided by superclass (initialize-instance :after index): use `backend-buffer' instead of `index-buffer' (close index): removed; provided by superclass (put-entry index integer integer integer): use `backend-buffer' instead of `index-buffer' (make-buffer index eql nil): new method; create `indx' instance (make-buffer index indx): new method; initialize `indx' instance (write-buffer index indx): do not reset state of `indx' instance; `make-buffer' method does that (buffer-property index indx eql :length/entries): new method; return number of entries (buffer-property index indx eql :length/bytes): new method; return size of index in bytes

Revision 60a0fb42 - 06/21/2012 02:28 AM - J. Moringen

Added last-write-time-mixin in src/backend/backend-mixins.lisp

refs #846, #847

- src/backend/backend-mixins.lisp: new file; contains mixin

classes for backend classes
- src/backend/package.lisp (package rsbag.backend): added exported
  symbol last-write-time-mixin * src/backend/tidelog/index.lisp (index): added superclass
  `last-write-time-mixin'
- src/backend/tidelog/file.lisp (file): likewise
- cl-rsbag.asd (system cl-rsbag): added file
  src/backend/backend-mixins.lisp

Revision 7fc11149 - 06/21/2012 03:28 AM - J. Moringen

Support flush strategies in bag-record/main.lisp
refs #846, #847

- bag-record/main.lisp (update-synopsis): added flush-strategy
  commandline option
  (main): pass value obtained from flush-strategy commandline option
  to `events->bag'
- bag-record/help.lisp (make-channel-strategy-help-string): minor
  improvement
  (make-flush-strategy-help-string): new function; return help
  string for flush strategies

Revision 47010ea7 - 06/21/2012 03:33 AM - J. Moringen

Added flush-strategy commandline option in bag-record.rst
fixes #846, #847

- bag-record.rst: added description of flush-strategy commandline
  option

History

#1 - 02/02/2012 03:53 PM - J. Moringen

- Status changed from New to Feedback

#2 - 02/03/2012 02:08 PM - J. Moringen

- Description updated

#3 - 02/04/2012 05:08 PM - I. Lütkebohle

Your interpretation is correct. While the spec does not explicitly say that multiple INDX blocks are possible, there is a tacit assumption that each block may occur multiple times, including INDX. I made a note to add this to the next draft.

#4 - 02/05/2012 01:25 PM - J. Moringen

- Status changed from Feedback to In Progress
- % Done changed from 0 to 20

Thanks for confirming this.

#5 - 03/06/2012 07:43 PM - J. Moringen

- Target version set to rsb-0.7

#6 - 06/21/2012 03:30 AM - J. Moringen

- % Done changed from 20 to 90

#7 - 06/21/2012 03:41 AM - J. Moringen

- Status changed from In Progress to Resolved

- % Done changed from 90 to 100

Applied in changeset r468.