

Robotics Service Bus - Feature #206

Enhancement # 548 (Resolved): Threadless RSB

Provide additional pull-based model for receiving events without threading overhead (Reader interface)

02/22/2011 05:22 PM - S. Wrede

Status:	Resolved	Start date:	02/22/2011
Priority:	Normal	Due date:	
Assignee:	J. Moringen	% Done:	100%
Category:	C++	Estimated time:	0.00 hour
Target version:	rsb-0.7		
Description			
<p>Result of brainstorming with Ingo due to our initialize experiences with mt-dispatching in Python on NAO:</p> <p>Currently, the EventDispatcher reads events from a Port, performs the matching, and dispatches matching events in multiple threads via callbacks to user code. This resembles coarsely to a <i>push</i>-based approach. Sometimes, it can be handy to follow a simpler <i>pull</i> pattern: Just provide an interface, which allows the user to receive events synchronously from a single thread on demand or blocking until a new event is available.</p> <p>The latter option was implemented in previous XCF versions with queues featuring receive or receiveLast methods, but there the queues itself were callbacks of the multi-threaded dispatching code. Now the idea would be to simplify things by implementing a pull-based event dispatcher that just directly puts the events in a queue without the overhead of multi-threaded dispatching.</p> <p>The rationale is that both use-cases are valid in different application-dependent contexts. Hence, we should support both. We should try this prototypically in RSBPython but eventually, this concept should be available also in Java and C++.</p>			
Related issues:			
Related to Robotics Service Bus - Tasks # 377: Pull-style Event Receiving		New	06/20/2011
Related to Robotics Service Bus - Tasks # 376: Pull-style Event Receiving		New	06/20/2011

Associated revisions

Revision d9ebf9e0 - 05/28/2011 08:11 PM - J. Moringen

Added initial infrastructure pieces for pull-style processing path
refs #206

More precisely, separated aspects which are specific to a push-style processing into specialized classes.

- src/CMakeLists.txt: added files
src/rsb/eventprocessing/PushEventReceivingStrategy.h and
src/rsb/eventprocessing/PushInRouteConfigurator.{h,cpp}
- src/rsb/Listener.{h,cpp}: create an object of
PushInRouteConfigurator instead of InRouteConfigurator
- src/rsb/eventprocessing/EventReceivingStrategy.h: removed methods
setHandlerErrorStrategy, addHandler and removeHandler since these
only make sense in a push-style data flow; methods are now in
src/rsb/eventprocessing/PushEventReceivingStrategy.h
- src/rsb/eventprocessing/PushEventReceivingStrategy.h: new file;
contains specialized PushEventReceivingStrategy class
- src/rsb/eventprocessing/ParallelEventReceivingStrategy.{h,cpp}:
changed base class to PushEventReceivingStrategy; fixed
qualification of class names and added an explanatory comment
- src/rsb/eventprocessing/InRouteConfigurator.{h,cpp}: removed methods
handlerAdded, handlerRemoved and setErrorStrategy since these only

make sense in a push-style data flow; methods are now in
src/rsb/eventprocessing/PushInRouteConfigurator.{h,cpp}; added
createEventReceivingStrategy to allow derived classes to provide
their on EventReceivingStrategy object

- src/rsb/eventprocessing/PushInRouteConfigurator.{h,cpp}: new file;
contains class PushInRouteConfigurator which is a specialization of
InRouteConfigurator for push-style data flows

Revision f1c48d39 - 06/20/2011 09:18 PM - J. Moringen

Added InPullConnector interface in src/rsb/transport/InPullConnector.{h,cpp}

refs #206

- src/CMakeLists.txt: added files
src/rsb/transport/InPullConnector.{h,cpp}
- src/rsb/transport/InPullConnector.{h,cpp}: new files; contain
InPullConnector interface for pull-style connectors

Revision 9e24c40b - 06/20/2011 09:57 PM - J. Moringen

Added class PullEventReceivingStrategy

refs #206

- src/CMakeLists.txt: added files
src/rsb/eventprocessing/PullEventReceivingStrategy.{h,cpp}
- src/rsb/eventprocessing/PullEventReceivingStrategy.{h,cpp}: new
files; contain class PullEventReceivingStrategy for pull-style event
receiving

Revision fb6f05ee - 06/20/2011 10:33 PM - J. Moringen

Added InPushConnector interface

refs #206

- src/CMakeLists.txt: added files
src/rsb/transport/InPushConnector.{h,cpp}
- src/rsb/eventprocessing/PushInRouteConfigurator.h: include
InPushConnector.h
- src/rsb/transport/InPushConnector.{h,cpp}: new files; contain
interface InPushConnector
- src/rsb/transport/inprocess/InConnector.{h,cpp}: changed base class
of inprocess::InConnector to InPushConnector; return InPushConnector
pointer from create method
- src/rsb/transport/spread/InConnector.{h,cpp}: changed base class of
spread::InConnector to InPushConnector; return InPushConnector
pointer from create method

Revision 6f97d466 - 06/20/2011 10:53 PM - J. Moringen

Added In{Pull,Push}Factory in src/rsb/transport/Factory.h

refs #206

- src/rsb/transport/Factory.h: added typedefs InPullFactory and
InPushFactory
- src/rsb/transport/transports.cpp: prepared setup of InPullFactory

Revision e4534b40 - 06/20/2011 11:06 PM - J. Moringen

Moved push-style-specific behavior to PushInRouteConfigurator

refs #206

- src/rsb/eventprocessing/InRouteConfigurator.cpp: removed adding the event receiving strategy to the list of handlers of the connectors
- src/rsb/eventprocessing/PushInRouteConfigurator.cpp: do it here instead

Revision 2f6e21e7 - 06/20/2011 11:19 PM - J. Moringen

Added PullInRouteConfigurator class

refs #206

- src/CMakeLists.txt: added files
src/rsb/eventprocessing/PullInRouteConfigurator.{h,cpp}
- src/rsb/eventprocessing/PullInRouteConfigurator.{h,cpp}: new files; contain the PullInRouteConfigurator class

Revision d174f005 - 06/20/2011 11:33 PM - J. Moringen

Removed push-style-specific in src/rsb/transport/InConnector.{h,cpp}

refs #206

- src/rsb/transport/InConnector.{h,cpp}: removed {add,remove}Handler methods; removed handler list data member; added virtual destructor; removed unused includes

Revision 4f2b88b3 - 06/20/2011 11:48 PM - J. Moringen

Use InPushConnector instead of InConnector in unit tests

refs #206

- test/rsb/transport/ConnectorTest.{h,cpp}: use InPushConnector instead of InConnector
- test/rsb/transport/inprocess/InProcessConnectorTest.cpp: likewise
- test/rsb/transport/spread/SpreadConnectorTest.cpp: likewise

Revision f496ce85 - 06/20/2011 11:56 PM - J. Moringen

Removed transport::InFactory typedef

refs #206

- apps/rsbversion/rsbversion.cpp: display InPushFactory and InPullFactory instead of InFactory
- src/rsb/Factory.{h,cpp}: moved connector creation from createListener and createInformer into new method createConnectors; this needs some meta-programming tricks :(
- src/rsb/Listener.{h,cpp}: expect a list of InPushConnectors in constructor
- src/rsb/transport/Factory.h: removed InFactory
- src/rsb/transport/transports.cpp: replaced InFactory with InPushFactory

Revision 0ca3c236 - 06/21/2011 12:10 AM - J. Moringen

Added Reader class

refs #206

- src/CMakeLists.txt: added files src/rsb/Reader.{h,cpp}
- src/rsb/Factory.{h,cpp}: added createReader method
- src/rsb/Reader.{h,cpp}: new files; contain Reader class

Revision 03a78600 - 06/21/2011 12:29 AM - J. Moringen

Added spread::InPullConnector class

refs #206

- src/CMakeLists.txt: added files
src/rsb/transport/spread/InPullConnector.{h,cpp}
- src/rsb/transport/transports.cpp: register Spread InPullConnector in InPullFactory
- src/rsb/transport/spread/InPullConnector.{h,cpp}: new files; contain spread::InPullConnector class

Revision d8db9cc3 - 06/21/2011 12:30 AM - J. Moringen

Added receive method to SpreadConnector class

refs #206

- src/rsb/transport/spread/SpreadConnector.cpp: added receive method

Revision fd35cde7 - 06/21/2011 12:36 AM - J. Moringen

Added example for Reader class

refs #206

- examples/CMakeLists.txt: added file examples/reader/reader.cpp
- examples/reader/reader.cpp: new file; contains example for Reader class

Revision 350ecd32 - 06/18/2012 03:05 PM - J. Moringen

Renamed transport::spread::{InConnector -> InPushConnector}

refs #206

- src/rsb/transport/spread/InConnector.{h,cpp}: renamed
src/rsb/transport/spread/{InConnector -> InPushConnector}.{h,cpp};
renamed transport::spread::{InConnector -> InPushConnector};
modified includes, forward decls, etc. accordingly
- src/rsb/transport/spread/InPullConnector.cpp: modified includes,
forward decls, etc. accordingly
- src/rsb/transport/spread/OutConnector.cpp: likewise
- src/rsb/transport/spread/ReceiverTask.{h,cpp}: likewise
- src/rsb/transport/spread/SpreadConnector.{h,cpp}: likewise
- src/CMakeLists.txt: renamed files
src/rsb/transport/spread/{InConnector -> InPushConnector}.{h,cpp}
- test/rsb/transport/ConnectorTest.{h,cpp}: likewise; distinguish
In{PushPull}Connector classes
- test/rsb/transport/spread/SpreadConnectorTest.cpp: likewise

Revision 9a8aa339 - 06/18/2012 05:35 PM - J. Moringen

Fixed spread::InPullConnector and related classes

fixes #206

- src/rsb/eventprocessing/PullEventReceivingStrategy.{h,cpp}: removed

- handle()-based processing
- src/rsb/transport/ConverterSelectingConnector.h: maybe typedefs public
- src/rsb/transport/transports.cpp: moved protocol buffer library version check here
- src/rsb/transport/spread/InPullConnector.{h,cpp}: use MessageHandler instance for message decoding
- src/rsb/transport/spread/MessageHandler.{h,cpp}: new files; implement push/pull-independent part of message decoding
- src/CMakeLists.txt: added files
src/rsb/transport/spread/MessageHandler.{h,cpp}

Revision 0f626a2a - 07/09/2012 05:54 PM - J. Moringen

Support reader interface in inprocess transport

refs #206

- src/rsb/transport/InConnector.h: use virtual inheritance; updated copyright
- src/rsb/transport/InPullConnector.h: likewise
- src/rsb/transport/InPushConnector.h: likewise
- src/rsb/transport/transports.cpp: register inprocess pull connector
- src/rsb/transport/inprocess/InConnector.{h,cpp}: new files; base-class for in process push and pull connectors
- src/rsb/transport/inprocess/Bus.{h,cpp}: use new InConnector base-class
- src/rsb/transport/inprocess/InPullConnector.{h,cpp}: moved some functionality to new InConnector class
- src/rsb/transport/inprocess/InPushConnector.{h,cpp}: likewise
- src/CMakeLists.txt: added files
src/rsb/transport/inprocess/InConnector.{h,cpp}

History

#1 - 04/29/2011 05:57 PM - J. Wienke

- *Category set to C++*

The discussed solution is to have a special handler with an internal queue and then block on this handler.

Moving this to C++ for the prototypical system modelling.

#2 - 04/29/2011 06:03 PM - I. Lütkebohle

Well, the "solution" is what I'm doing right now. It is certainly convenient to have framework support for that, but it doesn't change anything. Its just another level of indirection. What I asked for is to /remove/ the level of indirection that is currently there, which maps from the pull-based Spread API to RSB's API, and which I often don't need.

In other words, this ticket is not about functionality, its about removing complexity (and queueing twice, where none would suffice ;-).

#3 - 04/29/2011 06:06 PM - I. Lütkebohle

To be clear, I'm not asking that this be functionality be removed entirely. I just want it to be optional.

#4 - 04/29/2011 06:10 PM - J. Wienke

- Subject changed from *Refactoring of Event Dispatching Subsystem to Provide additional pull-based model for receiving events without threading overhead*

Ok, I remember the discussion.

#5 - 05/05/2011 06:07 PM - J. Wienke

Just a bit of brainstorming:

The current model for Listeners makes the transition to asynchronous notifications already on the lowest level of the transports. Hence, it does not make sense to provide the additional mode on the level. Instead I would propose to add a new Participant subclass called "Reader" which provides the pull-based model to get data from the bus. There will still be some modifications necessary on the transport level to support pulling there, too.

#6 - 05/05/2011 06:08 PM - I. Lütkebohle

Sounds like a good plan.

#7 - 05/13/2011 09:16 AM - J. Moringen

- Assignee set to J. Moringen

#8 - 05/20/2011 11:12 AM - J. Moringen

- Target version set to 0.3

We agreed on the following implementation strategy

- Extend backend (event processing + connectors) to support additional pull-path
 - This requires an additional connector class for each transport
- Add a client facing "Reader" class
- Limitations
 - If the reader accesses the bus using a single connector, a complete pull-based, queueing-free path is possible
 - If the reader accesses the bus using more than one connector, one layer of queueing will be required (until we implement a framework-level "select")

A prototype implementation (in which the Reader is called "receiver") is available in the Common Lisp RSB implementation.

#9 - 05/25/2011 09:37 PM - J. Moringen

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

- % Done changed from 0 to 30

#10 - 05/26/2011 03:19 PM - J. Wienke

- Target version deleted (0.3)

#11 - 06/20/2011 11:52 PM - J. Moringen

- Target version set to *rsb-0.10*

#12 - 06/21/2011 12:58 AM - J. Moringen

- % Done changed from 30 to 80

#13 - 10/20/2011 03:55 PM - J. Wienke

- Subject changed from *Provide additional pull-based model for receiving events without threading overhead* to *Provide additional pull-based model for receiving events without threading overhead (Reader interface)*

#14 - 10/20/2011 03:57 PM - J. Wienke

- Parent task set to #548

#15 - 06/18/2012 01:52 PM - J. Moringen

- Target version changed from *rsb-0.10* to *rsb-0.7*

#16 - 06/18/2012 05:35 PM - J. Moringen

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

- % Done changed from 80 to 100

Applied in changeset r3600.