Robotics Service Bus - Bug #1331
Parallel access to protobuf event payload not thread-safe in python?

01/15/2013 11:59 AM - D. Klotz

Status:	Resolved		**Start date:**	01/15/2013
Priority:	Normal		**Due date:**	
Assignee:	J. Wienke		**% Done:**	100%
Category:	Python		**Estimated time:**	0.00 hour
Target version:	rsb-0.7			

Description

We noticed a strange problem when both the logging code of rsb-python and my own handler trying to access the event payload object (which is a protobuf/RST type) in parallel, even though both should only be doing read access to the object. This is the exception, which has the unfortunate effect of stopping all further event delivery to my handler:

```
...
2013-01-15 11:50:51,909
/vol/humavips/releases/y3demo/lib/python2.7/site-packages/rsb_python-0.7.0-py2.7.egg/rsb/transport/socket/__init__.py 131
Receiving notification of size 167
2013-01-15 11:50:51,910
/vol/humavips/releases/y3demo/lib/python2.7/site-packages/rsb_python-0.7.0-py2.7.egg/rsb/transport/socket/__init__.py 155
Receive error: dictionary changed size during iteration
Traceback (most recent call last):
  File
"/vol/humavips/releases/y3demo/lib/python2.7/site-packages/rsb_python-0.7.0-py2.7.egg/rsb/transport/socket/__init__.py", line
151, in receiveNotifications
    self.doOneNotification()
  File
"/vol/humavips/releases/y3demo/lib/python2.7/site-packages/rsb_python-0.7.0-py2.7.egg/rsb/transport/socket/__init__.py", line
145, in doOneNotification
    self.dispatch(notification)
  File "/vol/humavips/releases/y3demo/lib/python2.7/site-packages/rsb_python-0.7.0-py2.7.egg/rsb/eventprocessing.py", line
76, in dispatch
    handler(event)
  File
"/vol/humavips/releases/y3demo/lib/python2.7/site-packages/rsb_python-0.7.0-py2.7.egg/rsb/transport/socket/__init__.py", line
272, in __call__
    self.handleIncoming(( connection, notification ))
  File
"/vol/humavips/releases/y3demo/lib/python2.7/site-packages/rsb_python-0.7.0-py2.7.egg/rsb/transport/socket/__init__.py", line
561, in handleIncoming
    super(BusServer, self).handleIncoming(connectionAndNotification)
  File
"/vol/humavips/releases/y3demo/lib/python2.7/site-packages/rsb_python-0.7.0-py2.7.egg/rsb/transport/socket/__init__.py", line
341, in handleIncoming
    self._toConnectors(notification)
  File
"/vol/humavips/releases/y3demo/lib/python2.7/site-packages/rsb_python-0.7.0-py2.7.egg/rsb/transport/socket/__init__.py", line
406, in _toConnectors
    connector.handle(notification)
  File
"/vol/humavips/releases/y3demo/lib/python2.7/site-packages/rsb_python-0.7.0-py2.7.egg/rsb/transport/socket/__init__.py", line
750, in handle
```

```
        self.__action(event)
    File "/vol/humavips/releases/y3demo/lib/python2.7/site-packages/rsb_python-0.7.0-py2.7.egg/rsb/eventprocessing.py", line
155, in handle
        self.__pool.push(event)
    File "/vol/humavips/releases/y3demo/lib/python2.7/site-packages/rsb_python-0.7.0-py2.7.egg/rsb/util.py", line 252, in push
        self.__logger.debug("Got new message to dispatch: %s" % message)
    File "/vol/humavips/releases/y3demo/lib/python2.7/site-packages/rsb_python-0.7.0-py2.7.egg/rsb/__init__.py", line 1068, in
__str__
        % ("Event", self.__id, self.__scope, printData, self.__type, self.__metaData, self.__causes)
    File "/usr/lib/python2.7/dist-packages/google/protobuf/internal/python_message.py", line 660, in __str__
        return text_format.MessageToString(self)
    File "/usr/lib/python2.7/dist-packages/google/protobuf/text_format.py", line 58, in MessageToString
        PrintMessage(message, out, as_utf8=as_utf8, as_one_line=as_one_line)
    File "/usr/lib/python2.7/dist-packages/google/protobuf/text_format.py", line 67, in PrintMessage
        for field, value in message.ListFields():
    File "/usr/lib/python2.7/dist-packages/google/protobuf/internal/python_message.py", line 554, in ListFields
        all_fields = [item for item in self._fields.iteritems() if _IsPresent(item)]
RuntimeError: dictionary changed size during iteration
```

Associated revisions

Revision 5eff49fa - 01/17/2013 02:09 PM - D. Klotz

Work around a bug in the protobuf implementation.

Without calling the *str* method of protobuf payloads, bug #1331 is
less likely to happen.

refs #1331

Revision 5e766071 - 01/17/2013 02:14 PM - D. Klotz

Backport: Work around a bug in the protobuf implementation.

Without calling the *str* method of protobuf payloads, bug #1331 is
less likely to happen.

refs #1331

History

#1 - 01/15/2013 02:06 PM - J. Wienke
- Status changed from New to In Progress

I posted to the mailing list to actually find out which thread safety guarantees exist in the python API. The documentation doesn't state anything.
Unfortunately, as a new member to the mailing list, my posting awaits moderation.

#2 - 01/16/2013 05:18 PM - D. Klotz
- File prototest.py added

So far I've unfortunately NOT been able to reproduce this error by just doing lots of reading operations on a ProtoBuf object from different threads (see the attached test case).

I'm also pretty sure that I'm not doing any write operations on the received data in my handler, so this is really confusing.

Also: Are we really sure that the payload is the only protobuf type contained in an rsb.Event (and its Metadata etc.) that is received by the handlers?

#3 - 01/16/2013 06:20 PM - J. Wienke

This also happens when using the spread transport. So no socket transport error.

#4 - 01/17/2013 09:25 AM - J. Wienke

From our discussion yesterday I just checked what happens to the event after passing it into the OrderedQueueDispatcherPool: nothing. So it must be the combination of your handler and the logger call in the pool.

#5 - 01/17/2013 02:06 PM - J. Wienke
- % Done changed from 0 to 30

This is an implementation issue in the protocol buffers python API. We have reported it there:
http://code.google.com/p/protobuf/issues/detail?id=454

#6 - 01/17/2013 02:21 PM - J. Wienke
- Status changed from In Progress to Resolved
- % Done changed from 30 to 100

We have tried to work around this issue with the attached commits on this report. By removing the logging output it is less likely that the bug will be triggered. However, it might still appear if multiple handlers are present and once invokes the iteration on the field dictionary while the other accesses an empty repeated field. We have to wait for a patch in protobuf to resolve this.

Files

prototest.py	1.58 KB	01/16/2013	D. Klotz