

An incredible journey on
catching random CORBA
timeouts



A photograph of a winter forest scene. The ground is covered in a thick layer of snow, with a faint path leading through the trees. The evergreen trees are heavily laden with snow, their branches drooping under the weight. The sky is a pale, overcast blue. The overall atmosphere is quiet and serene.

Spoiler: Not that random

Client side

boring – just a timeout:

```
94819 1575979206271 com.intellij.rt.execution.junit.JUnitStarter - PID=6601 94820
1575979206276 1575979206279
ERROR 10-12-2019 13:00:09.291 [main - f.s.t.e.RetriableTask] Caught exception, all
retries done for error: exception message is: IDL:Tango/DevFailed:1.0
Error Level 0:
  - desc: Device (development/test_server/0) timed out (>3000 ms)!
  - origin: development/test_server/0.class
fr.esrf.TangoApi.DeviceProxyDAODefaultImpl.read_attribute)
  - reason: org.omg.CORBA.TIMEOUT: client timeout reached vmcid: 0x0 minor
code: 0 completed: No
  - severity: ERROR

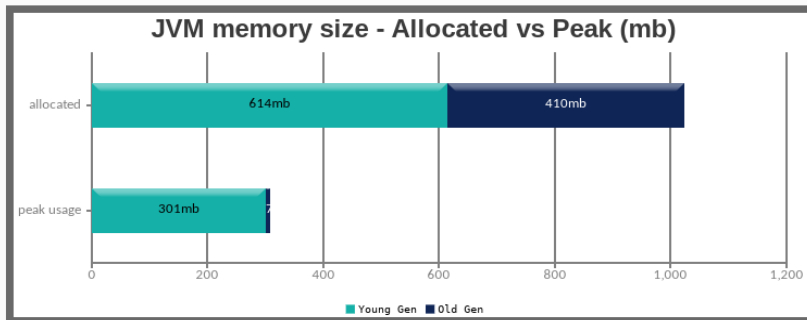
94820 1575979206279 timeout!!! 1575979209291
```



ServerSide

JVM memory size

Generation	Allocated 🔍	Peak 🔍
Young Generation	614 mb	301 mb
Old Generation	410 mb	7.5 mb
Total	1 gb	308.5 mb



Key Performance Indicators

(Important section of the report. To learn more about KPIs, [click here](#))

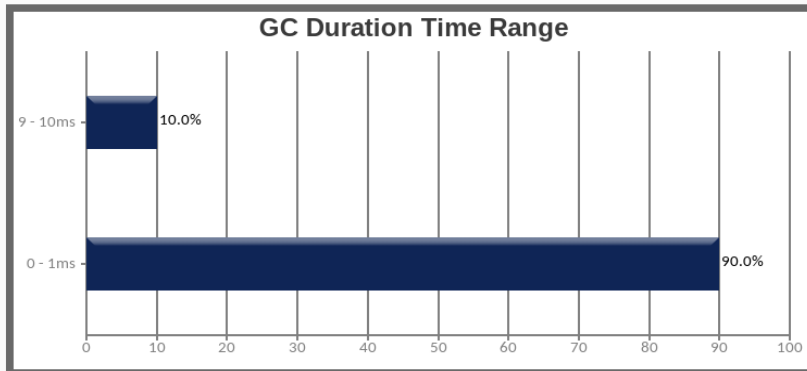
1 Throughput 🔍 : 99.957%

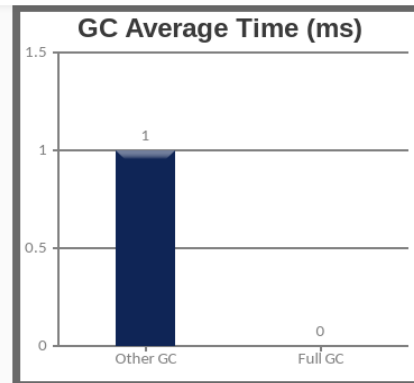
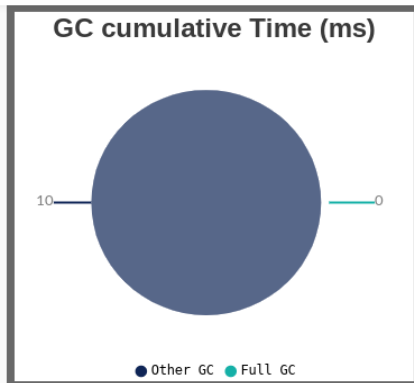
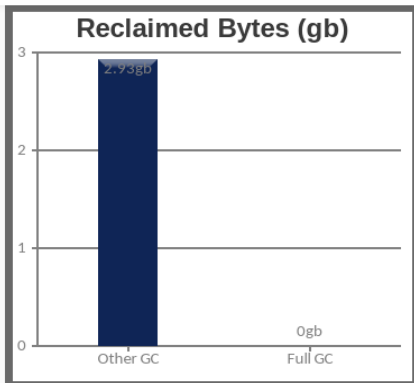
2 Latency:

Avg Pause GC Time 🔍	1.00 ms
Max Pause GC Time 🔍	10.0 ms

GC Pause Duration Time Range 🔍:

Duration (ms)		No. of GCs	Percentage
1	ms ▼ Change		
0 - 1		9	90.0%
9 - 10		1	10.0%





Total GC stats

Total GC count	10
Total reclaimed bytes	n/a
Total GC time	10.0 ms
Avg GC time	1.00 ms
GC avg time std dev	3.00 ms
GC min/max time	0 / 10.0 ms
GC Interval avg time	2 sec 580 ms

Other GC stats

Other GC count	10
Other GC reclaimed	2.93 gb
Other GC total time	10.0 ms
Other GC avg time	1.00 ms
Other GC avg time std dev	3.00 ms
Other GC min/max time	0 / 10.0 ms
Other GC Interval avg	2 sec 580 ms

Full GC stats

Full GC Count	0
Full GC reclaimed	n/a
Full GC total time	n/a
Full GC avg time	n/a
Full GC avg time std dev	n/a
Full GC min/max time	n/a
Full GC Interval avg	n/a

GC Pause Statistics

Pause Count	10
Pause total time	10.0 ms
Pause avg time	1.00 ms
Pause avg time std dev	0.0
Pause min/max time	0 / 10.0 ms

- Next stop - JacORB
 - logger debug mode not very useful
 - operation started/pause/invoke operation
 - Praise Open Source Software
 - download latest version
 - insert more logger
 - insert even more logger
 - insert even more than more logger

• Next stop - ServerRequestInterceptor

- line 85: `clientHostName.set(InetAddress.getByName(clientIPAddress.get()).getCanonicalHostName());`
- decompose insert more logger:

```
DEBUG 10-12-2019 13:00:06.282 [RequestProcessor-5 - o.j.o.p.AbstractInterceptorIterator] invoke( org.tango.orb.ServerRequestInterceptor )
TRACE 10-12-2019 13:00:06.282 [RequestProcessor-5 - o.t.o.ServerRequestInterceptor] infoImpl.getConnection().getTransport()
TRACE 10-12-2019 13:00:06.282 [RequestProcessor-5 - o.t.o.ServerRequestInterceptor] connex.getSocket()
TRACE 10-12-2019 13:00:06.282 [RequestProcessor-5 - o.t.o.ServerRequestInterceptor]
clientIPAddress.set(sock.getInetAddress().getHostAddress())
TRACE 10-12-2019 13:00:06.283 [RequestProcessor-5 - o.t.o.ServerRequestInterceptor] sock.getPort()
TRACE 10-12-2019 13:00:06.283 [RequestProcessor-5 - o.t.o.ServerRequestInterceptor]
InetAddress.getByName(clientIPAddress.get()).getCanonicalHostName()
TRACE 10-12-2019 13:00:06.283 [RequestProcessor-5 - o.t.o.ServerRequestInterceptor] clientIPAddress.get()
TRACE 10-12-2019 13:00:06.283 [RequestProcessor-5 - o.t.o.ServerRequestInterceptor] clientIPAddress.get() done: 131.169.65.104
TRACE 10-12-2019 13:00:06.283 [RequestProcessor-5 - o.t.o.ServerRequestInterceptor] InetAddress.getByName(clientIpAdress)
TRACE 10-12-2019 13:00:06.283 [RequestProcessor-5 - o.t.o.ServerRequestInterceptor] InetAddress.getByName(clientIpAdress) done:
/131.169.65.104
TRACE 10-12-2019 13:00:06.283 [RequestProcessor-5 - o.t.o.ServerRequestInterceptor] clientInetAddress.getCanonicalHostName()
!!! 5s delay !!!
TRACE 10-12-2019 13:00:11.288 [RequestProcessor-5 - o.t.o.ServerRequestInterceptor] clientInetAddress.getCanonicalHostName() done:
hzgc103k.desy.de
TRACE 10-12-2019 13:00:11.288 [RequestProcessor-5 - o.t.o.ServerRequestInterceptor] giopHostAddress.set(GIOP_TCP + clientIPAddress.get()
+ ":" + remotePort)
DEBUG 10-12-2019 13:00:11.288 [RequestProcessor-5 - o.j.o.p.AbstractInterceptorIterator] invoke( org.tango.orb.ServerRequestInterceptor )
done
```



```
clientInetAddress.getCanonicalHostName()
```

- JavaDoc:

- Gets the fully qualified domain name for this IP address. Best effort method, meaning we may not be able to return the FQDN depending on the underlying system configuration.

If there is a security manager, this method first calls its `checkConnect` method with the hostname and `-1` as its arguments to see if the calling code is allowed to know the hostname for this IP address, i.e., to connect to the host. If the operation is not allowed, it will return the textual representation of the IP address.

- Returns:

- the fully qualified domain name for this IP address, or if the operation is not allowed by the security check, the textual representation of the IP address.

- Since:

- 1.4

- So it makes kinda hidden call to DNS for each Tango request

- even worse – performance is unpredictable

Googling

- <https://github.com/netty/netty/issues/6454>
- <https://stackoverflow.com/questions/33289695/netaddress-getlocalhost-slow-to-run-30-seconds>
- https://bugs.eclipse.org/bugs/show_bug.cgi?id=153754
- etc

Most solutions

– enhance /etc/hosts:

- 127.0.0.1 localhost mbpro.local
- ::1 localhost mbpro.local

:(we do not want to manually set every host ip addr
in our network for every beamline

15 server/src/main/java/org/tango/orb/ServerRequestInterceptor.java

Viewed ...

```
55 - private volatile ThreadLocal<String> giopHostAddress = new ThreadLocal<String>();
56 - private volatile ThreadLocal<String> clientIPAddress = new ThreadLocal<String>();
53 + private final ThreadLocal<String> giopHostAddress = new ThreadLocal<String>();
54 + private final ThreadLocal<String> clientIPAddress = new ThreadLocal<String>();
57 55 private static final ServerRequestInterceptor INSTANCE = new ServerRequestInterceptor();
58 56
59 57 private ServerRequestInterceptor() {
@@ -82,15 +80,13 @@ public void receive_request(final ServerRequestInfo ri) throws ForwardRequest {
82 80 // remote informations
83 81 clientIPAddress.set(sock.getInetAddress().getHostAddress());
84 82 final int remotePort = sock.getPort();
85 - clientHostName.set(InetAddress.getByName(clientIPAddress.get()).getCanonicalHostName());
83 +
86 84 giopHostAddress.set(GIOP_TCP + clientIPAddress.get() + ":" + remotePort);
87 85 }
88 86 } else {
89 87 // when client is in the same process as the server, connection instance of
90 88 // org.jacorb.orb.iiop.IIOPListener$LoopbackAcceptor
91 - final InetAddress addr = InetAddress.getLocalHost();
92 - clientIPAddress.set(addr.getHostAddress());
93 - clientHostName.set(addr.getCanonicalHostName());
89 + clientIPAddress.set("127.0.0.1");
94 90 giopHostAddress.set(GIOP_TCP + clientIPAddress.get());
95 91 }
96 92 }
@@ -121,8 +117,9 @@ public String name() {
121 117 public void destroy() {
122 118 }
123 119
120 + @Deprecated
124 121 public String getClientHostName() {
125 - return clientHostName.get();
122 + return clientIPAddress.get();
126 123 }
127 124
128 125 public String getClientIPAddress() {
@@
```

```

82  @rest
83  @Ignore
84  public void testTimeoutAttribute() throws Exception{
85      TangoAttribute attr = new TangoAttribute( name: "development/test_server/0/testTimeoutAttribute");
86
87      for(long i = 0; i<1_000_000; ++i) {
88          long start = System.currentTimeMillis();
89
90          String result = null;
91          try {
92              result = attr.read(String.class);
93              System.out.println(String.format( s: "%d\t%d\t%s\t%d",i,start,result, System.currentTimeMillis()));
94          } catch (CommunicationTimeout timeout) {
95              System.out.println(String.format( s: "%d\t%d\t\ttimeout!!!\t%d",i,start,System.currentTimeMillis()));
96              throw timeout;
97          }
98      }
99  }
100 }
101

```

TestServerTest > testBenchmark_15s()

Run: TestServerTest.testTimeoutAttribute x TestServerTest.testBenchmark_15s x

Tests passed: 1 of 1 test - 5h 55m 49s 86ms

```

Ignored testTimeoutAttribute 5h 55m 49s 86ms
TestServerTest.testTimeoutA 5h 55m 49s 86ms
TestServerTest
DEBUG 10-12-2019 23:42:27.649 [main - o.j.o.e.StreamConnectionBase] wrote 84 bytes to 131.169.65.121:41055
DEBUG 10-12-2019 23:42:27.649 [main - o.j.o.e.StreamConnectionBase] wrote 52 bytes to 131.169.65.121:41055
DEBUG 10-12-2019 23:42:27.649 [main - o.j.o.e.StreamConnectionBase] wrote 120 bytes to 131.169.65.121:41055
DEBUG 10-12-2019 23:42:27.649 [main - o.j.o.g.GIOPConnection] wrote GIOP message of size 208 to ClientGIOPConnection to 131.169.65.121:41055 (29b5cd00)
DEBUG 10-12-2019 23:42:27.663 [ClientMessageReceptor1 - o.j.o.e.StreamConnectionBase] read 12 bytes from 131.169.65.121:41055
DEBUG 10-12-2019 23:42:27.663 [ClientMessageReceptor1 - o.j.o.g.GIOPConnection] read GIOP message of size 180 from 131.169.65.121:41055
DEBUG 10-12-2019 23:42:27.663 [ClientMessageReceptor1 - o.j.o.g.GIOPConnection] read GIOP message of size 192 from ClientGIOPConnection to 131.169.65.121:41055 (29b5cd00)
DEBUG 10-12-2019 23:42:27.663 [main - o.j.o.g.ClientConnectionManager] ClientConnectionManager: cannot release ClientGIOPConnection to 131.169.65.121:41055 (29b5cd00) (still has 3 cl
DEBUG 10-12-2019 23:42:27.663 [main - o.j.o.g.GIOPConnection] GIOPConnection.sendMessage timeout (millis): 3000
DEBUG 10-12-2019 23:42:27.664 [main - o.j.o.e.StreamConnectionBase] wrote 88 bytes to 131.169.65.121:41055
DEBUG 10-12-2019 23:42:27.664 [main - o.j.o.e.StreamConnectionBase] wrote 56 bytes to 131.169.65.121:41055
DEBUG 10-12-2019 23:42:27.664 [main - o.j.o.e.StreamConnectionBase] wrote 29 bytes to 131.169.65.121:41055
DEBUG 10-12-2019 23:42:27.664 [main - o.j.o.g.GIOPConnection] wrote GIOP message of size 125 to ClientGIOPConnection to 131.169.65.121:41055 (29b5cd00)
DEBUG 10-12-2019 23:42:27.674 [ClientMessageReceptor1 - o.j.o.e.StreamConnectionBase] read 12 bytes from 131.169.65.121:41055
DEBUG 10-12-2019 23:42:27.674 [ClientMessageReceptor1 - o.j.o.g.GIOPConnection] read GIOP message of size 288 from ClientGIOPConnection to 131.169.65.121:41055 (29b5cd00)
DEBUG 10-12-2019 23:42:27.675 [main - o.j.o.g.ClientConnectionManager] ClientConnectionManager: cannot release ClientGIOPConnection to 131.169.65.121:41055 (29b5cd00) (still has 3 cl
DEBUG 10-12-2019 23:42:27.675 [main - f.s.t.c.TangoAttribute] extracting fr.soleil.tango.clientapi.TangoAttribute@42607a4f[name=development/test_server/0/testTimeoutAttribute,type=De
999999 1576017747648 com.intellij.rt.execution.junit.JUnit4TestRunner - PID=1995 1000000 1576017747657 1576017747675

```

Process finished with exit code 0

A serene winter landscape featuring a snow-covered path that winds through a dense forest of evergreen trees. The trees are heavily laden with snow, their branches drooping under the weight. The ground is a smooth, white expanse, with a faint path leading the eye into the distance. The sky is a pale, clear blue, suggesting a bright but cold day. The overall atmosphere is peaceful and quiet.

Later on...

amend previous commit

 gwen-soleil committed on May 18 

84643f7



fix slowness that may occurs when getting host name, may fix #19

 gwen-soleil committed on May 18

43561ce



fix slowness that may occurs when getting host name, may fix #19

 gwen-soleil committed on May 18 

8b165a6



```
88 + // remote information
89 + String clientIPAddressTmp = sock.getInetAddress().getHostAddress();
90 + clientIPAddress.set(clientIPAddressTmp);
84 91 final int remotePort = sock.getPort();
85 - clientHostName.set(InetAddress.getByName(clientIPAddress.get()).getCanonicalHostName());
86 + giopHostAddress.set(GIOP_TCP + clientIPAddress.get() + ":" + remotePort);
92 + giopHostAddress.set(GIOP_TCP + clientIPAddressTmp + ":" + remotePort);
93 + // Workaround to resolve potential slowness of getCanonicalHostName()
94 + CompletableFuture<String> clientHostNameFutureTmp = CompletableFuture.supplyAsync(() -> {
95 +     String hostName;
96 +     try {
97 +         hostName = InetAddress.getByName(clientIPAddressTmp).getCanonicalHostName();
98 +     } catch (UnknownHostException e) {
99 +         // use IP address if unknown host
100 +         hostName = clientIPAddressTmp;
101 +     }
102 +     return hostName;
103 + });
104 + clientHostNameFuture.set(clientHostNameFutureTmp);
```

The end!



OpenSource + GitHub =



Refactor ZmqEventConsumer #20

Edit

Merged Ingvord merged 9 commits into [jtango-9-lts](#) from [issue-15](#) on Dec 4, 2019

Conversation 0

Commits 9

Checks 0

Files changed 14

+618 -2,243



Ingvord commented on Dec 4, 2019 • edited

Member + 😊 ...

- System.out(err) -> logger
- get rid of singletons and static methods
- do not implement TangoConsts
- do not implement Runnable
- [#18](#) Remove Notifd related code
- [#21](#) Merge EventConsumer and ZmqEventConsumer

Reviewers

No reviews

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Ingvord added 9 commits on Dec 4, 2019

- sout/serr -> logger c5027a4
- Progress [#15](#): remove NotifdEventConsumer 7bb98bc
- Progress [#15](#): inline EventConsumer into ZmqEventConsumer d56eb7a
- Progress [#15](#): ZmqEventConsumer clean up b442e4e
- Progress [#15](#): more Zmq
- Progress [#15](#): more Zmq

 [hzig-wpi/JTango/](#)