pyzmp Documentation

Release

Author

Apr 13, 2018

Contents

Contents:

CHAPTER 1

PyZMP

docs	
tests	
Python	
ROS	**Indigo Jade Kinetic**

PyZMP is a multiprocess library based on ZeroMQ.

The aim is to make experimenting with multiprocess and distributed architecture more solid and overall easier. If at all possible, the goal is to arrive at a minimal set of concepts, that makes solid and efficient distributed system easy to build.

Distributed systems models, as per wikipedia https://en.wikipedia.org/wiki/Distributed_computing#Models, can be classified as:

- Parallel algorithms in shared-memory model: This seems applicable in distributed software using a consensus algorithm as the shared memory.
- Parallel algorithms in message-passing model: This seems applicable in distributed software relying mostly on dataflow architecture, where the implementor can decide on the network structure
- Distributed algorithms in message-passing model: This seems to be the most widely used currently, (web back-end model, relying on services available from multiple places for example)

We will focus on the latter first, while keeping in mind it is likely just a special case of the second (network cannot be controlled, algorithm on each node has to be the same). A good exercise here is how to keep a representation of the distribution coherent on each node, despite potential network partition that cna occur.

Doing an Analysis on existing distributed software architecture is likely a very broad task, but we can focus on just a few here, at least as a first step. These should be enough to implement any of the distributed systems models cited above:

- service / RPC oriented architectures (usually implemented via https://en.wikipedia.org/wiki/Request%E2%80%93response)
- dataflow architecture (usually implemented via https://en.wikipedia.org/wiki/Publish%E2%80%93subscribe_pattern)

with different views for the user, more or less transparently : - make a request / send a task and (asynchronously or not) wait the response/result - receive dataflow from somewhere and send dataflow to somewhere else

Additionally, an interesting endeavour could be to see how https://en.wikipedia.org/wiki/Control_theory applies to such distributed systems (message passing <=> charge transfer).

Note : This is currently a personal perspective that likely require more thorough analysis, so feel free to send a Pull Request.

1.1 Repository structure

This repository has a few main branches:

- master : main branch, python dev workflow, releasing version tags into a pip package.

Apart from these we follow a feature branching workflow

1.2 How to use

Install ` `pip install pyzmp` `

Run self tests ` `pyzmp` `

1.3 How to develop

Clone this repository ` `git clone http://github.com/asmodehn/pyzmp` `

Create you virtualenv to workon using virtualenvwrapper ` `mkvirtualenv pyzmpenv` `

Install all dependencies via dev-requirements ` `pip install -r dev-requirements.txt` `

Run self tests ` `pyzmp` `

Run all tests (with all possible configurations) with tox ` `tox` `

Note : Tox envs are recreated every time to ensure consistency. So it s better to develop while in a non-tox-managed venv.

1.4 Tutorials and examples

A good example showing use of pyzmp, simple RPC client/server example

tutorials implementing multi node communication (under development)

1.5 Roadmap

Distributed software means software being executed in different "nodes" and collaboration via communication through different "channels":

- Node : A code executing entity. Can be a process, a thread, or a group of nodes communicating together.
- Channels : A way to make two or more node communicate in a way that allow them to collaborate.

This will allow us to structure our software in a network graph. PYZMP aims to be the foundation on which such a network graph can be easily, and confidently, built and used.

Implementation Priorities :

1. Local multiprocess first (we force data partition without forcing connections/sockets management)
2. Multiple concurrency implementation (Thread (all kinds), entity-component as a monothread implementation)
3. Remote concurrency (managing remote connections)

Type of Distributed Architecture that can be built with pyzmp:

1. Service(RPC) based architecture http://zguide.zeromq.org/page:all#Ask-and-Ye-Shall-Receive :

- It s a well proven way of architecture a distributed software, since it is the prevalent model used in the web architecture (REST, HTTP, RPC, etc.)
- There are some constraints in the way this must be implemented to work.
- There are more constraints if we want to implement it in a way that is easy to use.

2. DataFlow based architecture http://zguide.zeromq.org/page:all#Getting-the-Message-Out :

- It s a quite heavily used distributed architecture (topics, XMPP, ROS, etc.)
- There are some constraints in the way this must be implemented to work.
- There are more constraints if we want to implement it in a way that is easy to use.
- It is theoretically more complex to grasp than the service based architecture, therefore will be dealt with at a later time.

3. TBD : depending on analysis of existing system and what can be necessary to existing architecture, we will see what comes up.

Constraints:

- we want to be able to control where is executed what (no full transparency of the distribution)
- we want to create a solid platform on which other distributed algorithms can be implemented
- usual distributed algorithms (cache, proxy, feedback) should be super easy to implement, and will eventually be provided here as examples, or part of a larger "toolbox".
- We should minimize our software complexity on order to build a stable and easily maintainable system. A consensus algorithm (raft) would be very useful to implement distributed algorithms, but should be built outside of pyzmp. However pyzmp might need it to be able to function properly…

CHAPTER 2

pyzmp tutorial

2.1 Setup

First, we need to prepare the environment:

```
mkvirtualenv pyzmp
pip install .
```

2.2 Simple RPC client / server example

Here is an example using inheritance from pyzmp.Node:

```
(pyzmp)alexv@AlexV-Linux:~/Projects/pyzmp$ python
Python 2.7.6 (default, Jun 22 2015, 17:58:13)
[GCC 4.8.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import pyzmp
WARNING:root:ZMQ : Protobuf message implementation not found. Using pickle based␣
↪protocol
>>> class ServerNode(pyzmp.Node):
...     def __init__(self, name):
...             super(ServerNode, self).__init__(name)
...             self.the_answer= 42
...             self.provides(self.question)
...     def question(self):
...             return self.the_answer
...
>>> srv = ServerNode("srv")
>>> srv.start()
[srv] Node started as [6532 <= ipc:///tmp/zmp-srv-tVEvI_/services.pipe]
True
```

In another terminal you can overview the processes running:

```
alexv@AlexV-Linux:~$ pstree -cap 6532
python,6532
  ├─{python},6539
  └─{python},6540
```

Back into the same python intrepreter:

```
>>> question = pyzmp.discover("question")
>>> question
<pyzmp.service.Service object at 0x7fc1607215d0>
>>> question.call()
42
>>>
```

So here we have a very basic example of a communication between different processes through ZMQ.

2.3 Simple RPC client / server example with context management

A pyzmp Node is also a context manager (since there is a resource to initialize and cleanup : the process), which means you can shorten your code, and prevent leaving useless processes behind. The previous example can be rewritten

```
Python 2.7.12 (default, Dec  4 2017, 14:50:18)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import pyzmp
WARNING:root:ZMQ : Protobuf message implementation not found. Using pickle based␣
↪protocol
>>> class ServerNode(pyzmp.Node):
...     def __init__(self, name):
...             super(ServerNode, self).__init__(name)
...             self.the_answer = 42
...             self.provides(self.question)
...     def question(self):
...             return self.the_answer
...
>>> with ServerNode("srv") as srv:
...     question = pyzmp.discover("question")
...     question.call()
...
[srv] Proc started as [5393]
42
Shutdown initiated
>>>
```

Here the srv process was started when entering the "with:" block, and was terminated when exiting it.

2.4 Simple RPC client / server example with delegation

Often having hierarchy will make your code more complex, and difficult to evolve when requirements change. So you can also use pyzmp.Node as a delegate.

Disclaimer : currently BROKEN : see https://github.com/asmodehn/pyzmp/issues/36 You should implement your own context manager, which is what you want to do most of the time to control initialization anyway:

```
>>> class ServerNode(object):
...     def __init__(self, name):
...         self.node = pyzmp.Node(name)
...         self.the_answer = 42
...         self.node.provides(self.question)
...     def question(self):
...         return self.the_answer
...     def __enter__(self):
...         return self.node.__enter__()
...     def __exit__(self ,type, value, traceback):
...         return self.node.__exit__(type, value, traceback)
...
>>> with ServerNode("srv") as srv:
...     question = pyzmp.discover("question")
...     question.call()
...
[srv] Proc started as [5973]
Shutdown initiated
```

Context managers are used only in the child process, but enter() and exit() calls are in order which provide deterministic behavior, by contrast to multiprocess communication which is by default indeterministic.

A current limitation however is that discover currently works out of the box only from the same python interpreter. As a result we have to rely on a process manager running in the same interpreter.

A later version will provide an API to make this simple, even between two different interpreters, so that process management can be done somewhere else.

CHAPTER 3

pyzmp package

3.1 Subpackages

3.1.1 pyzmp.tests package

3.1.1.1 Submodules

3.1.1.2 pyzmp.tests.profile_node module

3.1.1.3 pyzmp.tests.test_message module

3.1.1.4 pyzmp.tests.test_node module

3.1.1.5 pyzmp.tests.test_service module

3.1.1.6 Module contents

3.2 Submodules

3.3 pyzmp.exceptions module

3.4 pyzmp.master module

3.5 pyzmp.message module

3.6 pyzmp.node module

3.7 pyzmp.service module

3.8 pyzmp.service_pb2 module

3.9 pyzmp.topic module

CHAPTER 4

Changelog

4.1 0.0.17 (2017-05-18)

- Removing quantified code badge. added gitchangelog config. [AlexV]
- Improved tutorial. [AlexV]
- Fixing coprocess test. [AlexV]
- Now passing arguments to context generators. [AlexV]
- Extracted coprocess from node. tests passing. [AlexV]
- Added test to ensure update is called once before we return control from start() call. [AlexV]
- Update twine from 1.8.1 to 1.11.0. [pyup-bot]
- Typo. [AlexV]

4.2 0.0.16 (2017-05-18)

- V0.0.16. [alexv]
- Signaling node start after checking for exit request, to guarantee at least one call to update. [alexv]
- Now signalling started before running target function. improved error handling on socket bind. [alexv]
- Bumping version since we have an API change. [alexv]
- Default node name is now uuid. now returning svc address on node start. [alexv]

4.3 0.0.15 (2017-05-02)

- Preparing 0.0.15. [AlexV]

- Update gitchangelog from 2.4.1 to 3.0.3. [pyup-bot]
- Now able to create daemon nodes. [alexv]
- Adding very basic node process watcher. [AlexV]
- Refining setproctitle usage. [AlexV]
- Small description change. [alexv]
- Importing modules for rosdevelop command only when that command is run so they are not dependencies for the python package itself. [alexv]
- Removing cmake lists from repo. to build among ros sources, use the rosdevelop command. [alexv]
- First version of rosdevelop setup.py command. [alexv]
- Fancy badges for README. [alexv]
- Fixing travis badge for ros release. [AlexV]
- Pin gitchangelog to latest version 2.4.1. [pyup-bot]
- Pin twine to latest version 1.8.1. [pyup-bot]
- Integrating setproctitle inside the node.run() method. [AlexV]
- Added reference to pyzmp tutorials in readme. [dhiraj dhule]
- Cleaned up node logs. fixed node update test. [alexv]
- Adding badge for ROS release build. [alexv]

4.4 0.0.14 (2016-08-10)

- Preparing 0.0.14. [alexv]
- Removing release shell script. now done from setup.py. [alexv]
- Removing package.xml. doing thirdparty release from release repo now. [alexv]
- Moving docs to doc. [alexv]
- Added doc-requirements. [AlexV]
- Added tutorial and example changelog generated with gitchangelog. [AlexV]
- More docs about process managers… [AlexV]
- Added very basic rpc tutorial. [AlexV]
- First doc version generated with sphinx-apidoc. added CHANGELOG. [AlexV]

4.5 0.0.13 (2016-08-10)

- Preparing 0.0.13. [alexv]
- Retrieving changes from indigo-devel, because it s building now and ready to be offloaded to release repo as patches. [alexv]
- Improved setup.py. now relying on twine for release. [alexv]
- Preparing version 0.0.12. [alexv]

4.6 0.0.12 (2016-08-09)

- Preparing version 0.0.12. [alexv]
- Synchronizing package version for ROS. [alexv]
- Improving self test. [AlexV]
- Refining tox venvs to not use them for development. [AlexV]
- Reviewing tox and tests. [AlexV]
- Filling in README. [AlexV]

4.7 0.0.11 (2016-06-23)

- Preparing version 0.0.11. [alexv]
- Listing pytest last in th dependency list to solve https://github.com /pytest-dev/pytest/issues/1652. [alexv]

4.8 0.0.10 (2016-06-23)

- Preparing version 0.0.10. [alexv]
- Documenting test methods to make quantifiedcode happy. [alexv]
- Updated quantifiedcode badge to point to correct project. [alexv]
- Return exitcode 0 if update ran, but never returned one. made run() method overridable more or less like Process.run() fixed confusion between run&update arguments cosmetics. [alexv]
- Change on API for shutdown and run/target. now process a lazy delegate to allow easy restart. [AlexV]
- Small node improvements. WIP. [AlexV]
- Reviewing how we use zmp nodes and improving tests… WIP. [alexv]
- Making default behavior (call start() without args) bw compatible (we don't know if node has started just yet) [alexv]
- Making default timeout infinite for start() and shutdown() just like multiprocessing.Process. [alexv]
- Waiting for node to be initialized before returning from start() call. made start() more determinist, tests more stricts, and added timeouts on start() and shutdown() [alexv]
- Removing site-packages since this is a pure python project. ROS test somewhere else. [alexv]
- Revert "travis with sudo to be able to install ros." [alexv]

  This reverts commit 7b59cdf84f4e83a8cb0f2c3242e64667d29323da.
- Revert "adding ROS install for travis tests" [alexv]

  This reverts commit 518fdadeca719e64140814b39a3c185b6da649d7.
- Travis with sudo to be able to install ros. [alexv]
- Calling py.test directly from tox. [alexv]
- Adding ROS install for travis tests. [alexv]
- Added files required by catkin-pip. [alexv]

- Now using pytest for testing. [alexv]
- Fixing release script tag command. [alexv]

4.9 0.0.9 (2016-05-04)

- Removed bloom from release script. tested with testpypi. [alexv]
- Small travis improvement. preparing v0.0.9. [alexv]
- Now using pytest and tox for testing. [alexv]
- Removing package.xml from manifest. [alexv]
- Adding basic travis first config. [alexv]
- Importing zmp from pyros, cleaning up pyros stuff. [alexv]
- Replacing obsolete navi/semantic_locations by new /rocon/semantics/locations. [alexv]
- Moved pyros and zmp sources, otherwise pyros was not find through egg link. [alexv]
- Added version. fixed tests in cmakelists. added default config file, removed useless testing config. added entry point for selftests. added requirements devel dependency to pyros-setup. [alexv]
- Cleaning up rosinterface __init__. now doing ros setup only in child node process, dynamically. parent process is isolated. [alexv]
- Cleaning up imports and fixing tests. [alexv]
- Refactored to add configuration at module, package and user levels. implified pyros-setup configuration from rosinterface. reviewed separation between node and interface to isolate all ros setup in child process. now doing ROS message conversion internally in rosinterface service and topic classes. fixed most tests. now uses six to improve python3 compatibility. [alexv]
- Starting to adapt to new configuration from pyros-setup. [alexv]
- Now using catkin_pure_python. [alexv]
- Add Gitter badge. [The Gitter Badger]
- Cosmetics, comments and small fixes… [alexv]
- Readme regarding IoT. [alexv]
- Cosmetics. [alexv]
- Changing reinit method to a setup service. now reinitialize rosinterface everytime the list of services or topic passed by the user changes. refactor the base interface to hold local copy of system state. fix all tests. [alexv]
- Added missing rosservice dependency. [alexv]
- Fixing package dependencies for catkin. [alexv]
- Fixing catkin build. [alexv]
- Removing unused ROS service specifications. [alexv]
- Improved exception handling. adding mock client to make unittests easy. cosmetics. [alexv]
- Improved Readme. [AlexV]
- Removing dynamic_reconfigure. [alexv]
- Removed rocon feature. cleanup. [alexv]

- Exposing servicecall timeout exception. cosmetics. [alexv]
- Warn -> info when it's not meant to be alarming to the users. [Daniel Stonier]
- Fixing log warn -> info for startup args. [alexv]
- Fixme comments. [alexv]
- Adding simple test to assert rospy potentially strange behaviors. separating cache and non cache tests. catching connection_cache proxy init timeout, showing error and disabling. [alexv]
- Adding custom manager argument in basenode, and making shutdown possible override more obvious. [alexv]
- ZMP : services and node advertisement now done in context managers. Node now support using custom context manager when starting in another process. cosmetics. [alexv]
- Improving base support to pass diff instead of query full state everytime. now with callback called from connection cache proxy to only process list if change happens. [alexv]
- Fixing reinit to be delayed if ros interface not ready yet. [alexv]
- Fixing pyrosROS test with latest pyros_test. [alexv]
- Adding pyrosRos test to catkin tests. [alexv]
- Reiniting connection cache if dynamic_reconfigure disable/enable it. [alexv]
- Using enable_cache in dynamic_reconfigure to be able to dynamically switch if needed. [alexv]
- Fixed populating empty message instance. comments. [alexv]
- Adding missing rosnode as test dependency. [AlexV]
- Disabling roconinterface dynamic import. [AlexV]
- Moving more nodes to pyros-test. [AlexV]
- Moving nodes to pyros-test. skipping tests if connection_cache not found. [AlexV]
- Better error message if tests are run from python without pyros-test installed in ROS env. [AlexV]
- Using pyros_cfg and fix import in rocont interface, to run nosetests from python venv. [AlexV]
- Added generated code for dynamic_reconfigure. [AlexV]
- Adding requirements, fixing setup.py for setuptools. [AlexV]
- Now allowing to delay the import of rosinterface subpackage and passing base_path to find ROS environment dynamically. [alexv]
- Using ros-shadow-fixed for travis. [AlexV]
- Cleaning up comments. [alexv]
- Adding option to enable cache or not from rosparams. [alexv]
- Ros_interface now using topics and service types from cacche if available, otherwise query one by one when needed. making sure cache process is started and stopped during the test to avoid scary harmless warnings. [alexv]
- Improving tests. [alexv]
- Using silent fallback for connectioncache proxy. [alexv]
- Fixing dependencies in package.xml. [alexv]
- Pyros now dependein on pyros_setup and pyros_test for tests. [alexv]
- Pyros now depending on pyros_setup. [alexv]

- Expose_transients_regex now relying on _transient_change_detect directly. small refactor to allow transient updates only with ROS system state differences. fixing mockinterface to call reinit only after setting up mock Added first connection_cache subscriber implementation to avoid pinging the master too often. WIP. [alexv]

4.10 0.0.8 (2016-01-25)

- Doing zmp tests one by one to workaround nose hanging bug with option –with-xunit. [alexv]
- Making service and param new style classes. [alexv]
- Fixing throttling to reinitialize last_update in basenode. [alexv]
- Fixing a few quantifiedcode issues… [alexv]
- ZMP node now passing timedelta to update. Pyros nodes now have a throttled_update method to control when heavy computation will be executed (potentially not every update) [alexv]
- Displaying name of ROS node in log when starting up. [alexv]
- Mentioning dropping actions support in changelog. [alexv]
- Overhauled documentation. [alexv]
- Cosmetics. [alexv]
- Exposing pyros service exceptions for import. [alexv]
- Adding node with mute publisher for tests. [alexv]
- Fixing basic test nodes return message type. cosmetics. [alexv]
- Reviewing README. [alexv]
- Changelog for 0.1.0. cosmetics. [alexv]
- Migrated % string formating. [Cody]
- Fixing badges after rename. [alexv]
- Avoid mutable default arguments. [Cody]
- Made namedtuple fields optional like for protobuf protocol. [alexv]
- Fixing zmp tests with namedtuple protocol. [alexv]
- Fixing catkin cmakelists after test rename. [alexv]
- Making client exceptions also PyrosExceptions. [alexv]
- Begining of implementation of slowservice node for test. not included in tests yet. [alexv]
- Removed useless hack in travis cmds, fixed typo. [alexv]
- Trying quick hack to fix travis build. [alexv]
- Adding status message when creating linksto access catkin generated python modules. [alexv]
- Adding zmp tests to catkin cmakelists. [alexv]
- Added dummy file to fix catkin install. [alexv]
- Small install and deps fixes. [alexv]
- Simplifying traceback response code in node. [alexv]
- Fixing unusable traceback usecase in zmp. [alexv]

- Cosmetics. adding basemsg unused yet. [alexv]
- Moving exception to base package, as they should be usable by the client of this package. [alexv]
- Making pyros exceptions pickleable. minor fixes to ensure exception propagation. [alexv]
- Comments. [alexv]
- Ros_setup now use of install workspace optional. fixes problems running nodes (which needs message types) from nosetests. [alexv]
- Added cleanup methods for transients. it comes in handy sometime (for ROS topics for example). [alexv]
- Pretty print dynamic reconfigure request. [alexv]
- Cleanup debug logging. [alexv]
- Adding logic on name was not a good idea. breaks underlying systems relaying on node name like params for ROS. [alexv]
- Removing name from argv, catching keyboard interrupt from pyros ros node. cosmetics. [alexv]
- Increasing default timeouts for listing services call form pyros client. [alexv]
- Fixed multiprocess mutli pyros conflict issues with topics with well known rosparam. now enforcing first part of node name. cosmetics. [alexv]
- Removed useless logging. [alexv]
- Adding basetopic and fixed topic detection in rosinterface. zmp service now excepting on timeout. [alexv]
- Fixed exceptions handling and transfer. fixed serialization of services and topic classes for ROSinterface. [alexv]
- Now reraise when transient type resolving or transient instance building fails. added reinit methods to list of node service to be able to change configuration without restarting the node (usecase : dynamic reconfigure) added option to PyrosROS node to start without dynamic reconfigure (useful for tests and explicit reinit) added some PyrosROS tests to check dynamic exposing of topics. cleaned up old rostful definitions. cosmetics. [alexv]
- Cleaning up old action-related code. fixed mores tests. [alexv]
- Fixing how to get topics and services list. commented some useless services (interactions, ationcs, etc.). [alexv]
- Changing version number to 0.1.0. preparing for minor release. [alexv]
- Refactoring ros emulated setup. [alexv]
- Improving and fixing rosinterface tests. still too many failures with rostest. [alexv]
- Fixing tests for Pyros client, and fixed Pyros client discovery logic. cosmetics. [alexv]
- Making RosInterface a child of BaseInterface and getting all Topic and test services to pass. cosmetics. [alexv]
- Improved test structure for rostest and nose to collaborate… [alexv]
- WIP. reorganising tests, moved inside package, nose import makes it easy. still having problems with rostest. [alexv]
- Fixing testTopic for rostest and nose. cosmetics. [alexv]
- Finishing python package rename. [alexv]
- Separated rospy / py trick from test. [alexv]
- Fixing testRosInterface rostest to be runnable from python directly, and debuggable in IDE, by emulating ROS setup in testfile. [alexv]
- Implemented functional API, abstract base interface class, mockinterface tests. [alexv]

- Moving and fixing tests. [alexv]
- Changing ros package name after repository rename. [alexv]
- Fixing setup.py for recent catkin. [alexv]
- Protecting rospy from unicode args list. [alexv]
- Implemented transferring exception information via protobuf msg. readding tblib as dependency required for trusty. [alexv]
- WIP. starting to change message to be able to just not send the traceback if tblib not found. [alexv]
- Restructuring code and fixing all tests to run with new zmp-based implementation. [alexv]
- Now able to use bound methods as services. [alexv]
- Adding python-tblib as catkin dependency. [alexv]
- Useful todo comments. [alexv]
- Now using pickle is enough for serialization. getting rid of extra dill and funcsig dependencies. [alexv]
- Not transmitting function signature anymore. not needed for python style function matching. [alexv]
- Added cloudpickle in possible serializer comments. [alexv]
- Now forwarding all exceptions in service call on node fixed all zmp tests. [alexv]
- Fixing all zmp tests since we changed request into args and kwargs. [alexv]
- Starting to use dill for serializing functions and params. [alexv]
- Adding comments with more serialization lib candidates… [alexv]
- WIP. looking for a way to enforce arguments type when calling a service, and parsing properly when returning an error upon exception. [alexv]
- Getting message to work for both protobuf and pickle. Now we need to choose between tblib and dill for exception serialization. [alexv]
- Adding dill as dependency. [alexv]
- Multiprocess simple framework as separate zmp package. [alexv]
- Comments. [alexv]
- Transferring exceptions between processes. [alexv]
- Fixing all service tests and deadlock gone. [alexv]
- Improved service and node tests. still deadlock sometimes… [alexv]
- Multiprocess service testing okay for discover. [alexv]
- WIP. starting to use zmq for messaging. simpler than other alternatives. [alexv]
- WIP implementing service. [alexv]
- WIP adding mockframework a multiprocess communication framework. [alexv]
- Adding mockparam. [alexv]
- Adding code health badge. [alexv]
- Adding requirements badge. [alexv]
- Adding code quality badge. [alexv]
- Adding echo tests for mocktopic and mockservice. [alexv]

- Renaming populate / extract commands. [alexv]
- Setting up custom message type and tests for mock interface. [alexv]
- Fixing mockmessage and test. [alexv]
- Improving mockmessage and tests. [alexv]
- Started to build a mock interface, using python types as messages. This should help more accurate testing with mock. [alexv]
- Adding six submodule. tblib might need it. otherwise it might come in useful anyway. [alexv]
- Adding tblib to be able to transfer exception between processes. [alexv]
- Fixing travis badge. [alexv]
- Adding travis badge. [alexv]
- Starting travis integration for autotest. [alexv]
- Adding rostopic as a test_depend. [alexv]
- Fixes to make this node work again with rostful cosmetics and cleanups. [alexv]
- First implementation to expose params to python the same way as we do for topics and services. [alexv]

4.11 0.0.7 (2015-10-12)

- 0.0.7. [alexv]
- Adding log to show rostful node process finishing. [alexv]
- Change message content check to accept empty dicts. [Michal Staniaszek]
- Fixing corner cases when passing None as message content. invalid and should not work. [alexv]
- Fixing tests. and changed api a little. [alexv]
- Removing useless fancy checks to force disabling rocon when set to false. updated rapp_watcher not working anymore. [AlexV]
- Rocon_std_msgs changed from PlatformInfo.uri to MasterInfo.rocon_uri. [AlexV]
- Send empty dicts instead of none from client. [Michal Staniaszek]
- Service and topic exceptions caught and messages displayed. [Michal Staniaszek]
- Fleshed out topic and service info tuples. [Michal Staniaszek]
- Can check for rocon interface, get interactions. [Michal Staniaszek]
- Listing functions for client, corresponding mock and node functions. [Michal Staniaszek]
- Now passing stop_event as an argument to the spinner. cosmetics. [alexv]
- Fix when running actual rostfulnode. [alexv]
- Now running rostful_node in an separate process to avoid problems because of rospy.init_node tricks. [alexv]
- Cosmetics. [alexv]
- Improving how to launch rostest test. fixed hanging nosetest. hooking up new test to catkin. [alexv]
- Force-delete for services, test for removal crash on expose. [Michal Staniaszek]

  Test service nodes added

- Fix crash when reconfigure removes topics, started on unit tests. [Michal Staniaszek]
- Fixing removing from dictionary topic_args. [alexv]
- Stopped removal of slashes from front of topics. [Michal Staniaszek]
- Fixed regex and add/remove issues with topics and services. [Michal Staniaszek]
- Fixed topic deletion, multiple calls to add. [Michal Staniaszek]

  The interface now tracks how many calls have been made to the add function and ensures that topics are not prematurely deleted from the list. Actions also have a similar thing going on, but not sure if it works since they are unused. Services are unchanged.

  Ensured uniqueness of topics and services being passed into the system using sets.

  Removed unnecessary ws_name code.

  Issue #27.
- Fix *_waiting list usage, service loss no longer permanent. [Michal Staniaszek]

  The lists *_waiting now contain topics, services or actions which we are expecting, but do not currently exist. Once it comes into existence, we remove it from this list.

  When services disconnect, their loss is no longer permanent. This had to do with the services being removed and not added to the waiting list.

  Fixes issue #21.
- Full regex, fixed reconfigure crash. [Michal Staniaszek]

  Can now use full regex in topic or service strings to match incoming strings.

  Fixed crash when dynamic reconfigure receives an invalid string
- Strings with no match characters don't add unwanted topics. [Michal Staniaszek]

  Regex fixed with beginning and end of line expected, previously would allow a match anywhere in the string.

  Issue #17.
- Removed separate lists for match strings. [Michal Staniaszek]
- Remove printing, unnecessary adding to _args arrays. [Michal Staniaszek]
- Adding wildcard * for exposing topics or services. [Michal Staniaszek]

  Implementation should be such that other match characters can be easily added if necessary.

  Fixes issue #17.
- Added TODO. [alexv]
- Added exception catching for when rocon interface is not available. [Michal Staniaszek]
- Added important technical TODO. [alexv]
- Fixing bad merge. [alexv]
- Fixing unitests after merge. [AlexV]
- Quick fix to keep disappeared topics around, waiting, in case they come back up… [alexv]
- Turning off consume/noloss behavior. should not be the default. should be in parameter another way to expose topics. [AlexV]
- Allowing to call a service without any request. same as empty request. [AlexV]
- Keeping topics alive even after they disappear, until all messages have been read… WIP. [AlexV]

- Preparing for release 0.0.6. setup also possible without catkin. [AlexV]
- Changing rostful node design to match mock design. [AlexV]
- Fixing RostfulCtx with new Mock design. added unittest file. [AlexV]
- Improved interface of rostful client. added unit tests for rostfulClient. [AlexV]
- Improved interface of rostful mock, now async_spin return the pipe connection. added more unit tests for rostful mock. [AlexV]
- Added rostful mock object (useful if no ROS found). improved structure and added small unit test. [AlexV]
- Changing cfg file name to fix install. [AlexV]
- Comments TODO to remember to fix hack. [AlexV]
- Tentative fix of cfg… comments. [AlexV]
- Adding python futures as dependency. [AlexV]
- Commenting out icon image. no cache home on robot. need to find a new strategy. [AlexV]
- Removed useless broken services. [AlexV]
- Fixing catkin_make install with dynamic reconfigure. [AlexV]
- Adding bloom release in release process to sync with pypi release. [AlexV]
- Fixes for release and cosmetics. [AlexV]
- Preparing pypi release. [AlexV]
- Improving rostful node API. Adding rostful pipe client and python pipe protocol. removed redundant ros services. [AlexV]
- Simplifying rapp start and stop by using rapp_watcher methods. [AlexV]
- Now starting and stopping rapp. still ugly. [AlexV]
- Fixes to get rocon features to work again. [AlexV]

4.12 0.0.3 (2015-07-01)

- Preparing pypi release. small fix. [AlexV]
- Adding helper services to access Rosful node from a different process. Hacky, working around a limitation of rospy (cannot publish on a topic created in a different process for some reason…). Proper design would be to call directly the python method (work with services - node_init not needed) [AlexV]
- Small cleanup. [AlexV]
- Adding context manager for rospy.init_node and rospy.signal_shutdown. No ROS signal handlers anymore. Cleanup properly done when program interrupted. [AlexV]
- Playing with signal handlers… [AlexV]
- Improved test. but topic interface not symmetric. needs to deeply test message conversion. [AlexV]
- Small fixes and first working test to plug on existing topic. [AlexV]
- Adding first copy from rostful. splitting repo in 2. [AlexV]
- Initial commit. [AlexV]

CHAPTER 5

Indices and tables

- genindex
- modindex
- search