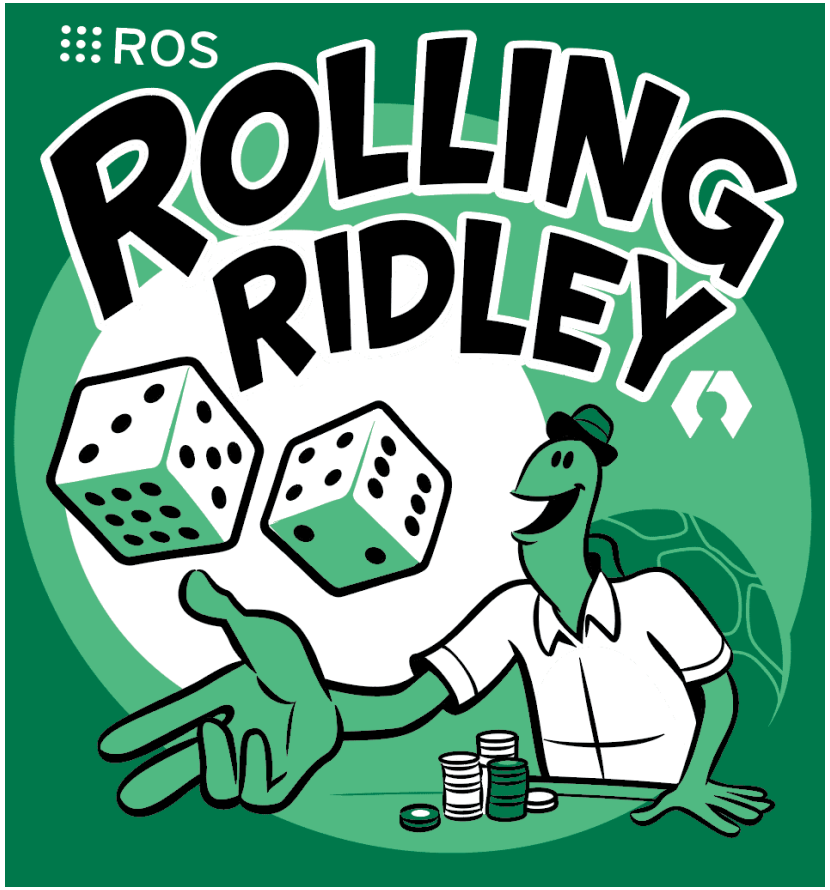
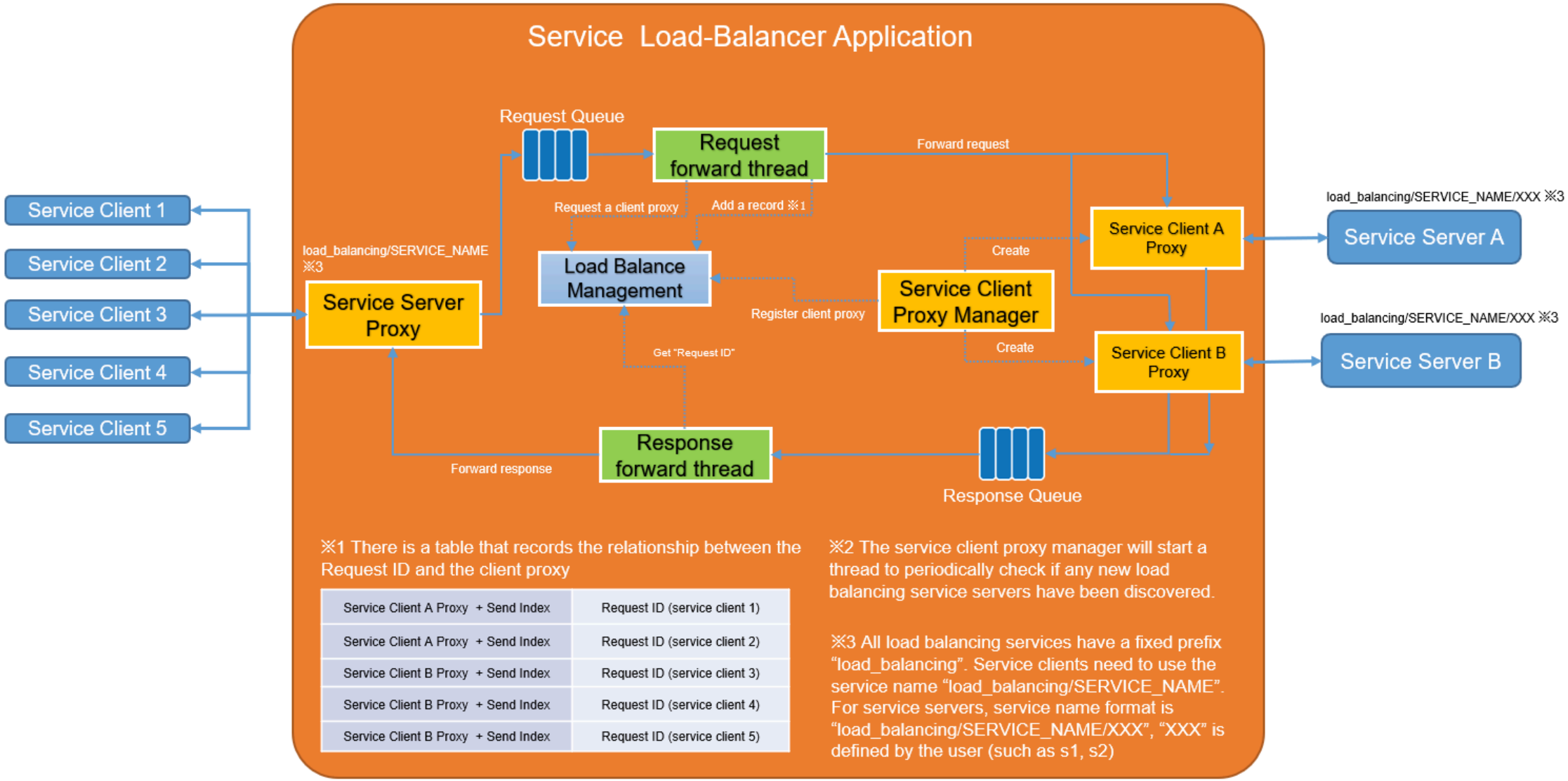


ROS2 Load Balancing Service

Objective

- Support multiple service servers on the same service path to implement redundancy and load-balancing.
- Existing ROS2 service server/client programs can be used without code modification.



- Service Server Proxy is implemented based on rclcpp::GenericService.
- Service Client Proxy is implemented based on rclcpp::GenericClient.

How to use

```
Usage:
    load_balancing_service [-h|--help] [-s|--service-name SERVICE_NAME] [-t|--service-type SERVICE_TYPE] [--strategy XXX] [-i|--interval TIME]
       --strategy choose one of "round_robin", "less_requests" and "less_response_time"
                  If not set, "round_robin" is used as default.
                  "round_robin": select the service server in order.
                  "less_requests": select the service server with the fewest requests.
                  "less_response_time": select the service server with the shortest average response time.
       --interval Interval to discovery service servers. Unit is second.
                  If not set, default is 1s.
```

The following parameters are required to start the load balancing service application.

- Service name
  This is the original service name. The service server proxy adds a fixed prefix "load_balancing", so the proxy service name becomes "load_balancing/SERVICE_NAME".
- Service Type
  such as "example_interfaces/srv/AddTwoInts"
- Strategy [Optional]
  The strategy for load balancing. Currently, 3 strategy modes are supported. They are "round_robin", "less_requests" and "less_response_time". The default strategy is round_robin.
- Interval [Optional]
  This parameter sets how often the service server discovery action is performed. The default interval is 1 second.

An example

Use demo https://github.com/ros2/demos/tree/rolling/demo_nodes_cpp/src/services as an example.

At first, run load balancing service application in a terminal

```
$ ros2 run load_balancing_service load_balancing_service -s add_two_ints -t example_interfaces/srv/AddTwoInts --strategy round_robin -i 1
[INFO] [1727418589.343051995] [main]:
   Load balancing service name: /load_balancing/add_two_ints
                  Service type: example_interfaces/srv/AddTwoInts
       Load balancing strategy: round_robin
  Interval to discovery server: 1s
------------------------------
Service client remap service name to /load_balancing/add_two_ints
Service server remap service name to /load_balancing/add_two_ints/XXX
```

The output log will provide hints for the service names that service clients and service servers need to use.

- For service client, it should remap service name to "/load_balancing/add_two_ints".
- For service server, it should remap service name to "/load_balancing/add_two_ints/XXX". "XXX" is specified by user. Such as "/load_balancing/add_two_ints/s1".

Run 2 service servers

Open a terminal, run the below command

```
$ ros2 run demo_nodes_cpp add_two_ints_server --ros-args -r add_two_ints:=load_balancing/add_two_ints/s1
```

Open another terminal, run the below command

```
$ ros2 run demo_nodes_cpp add_two_ints_server --ros-args -r add_two_ints:=load_balancing/add_two_ints/s2
```

Run 10 service clients

Open another terminal, run the this script

```
$ cat run_clients.sh
#!/bin/bash

for i in {1..10}
do
  ros2 run demo_nodes_cpp add_two_ints_client_async --ros-args -r add_two_ints:=load_balancing/add_two_ints &
done
```

Eventually, you will see logs that 5 requests have been received in the two terminals running the service server.

Project repository

https://github.com/barry-Xu-2018/ros2_load_balancing_service/