4. Multi-computer communication configuration

4.1. Concept

Multi-computer communication, or distributed communication, is a communication strategy that can realize data interaction between different hosts through the network.

ROS2 itself is a distributed communication framework, which can conveniently realize the communication between different devices. The middleware based on ROS2 is DDS. When in the same network, distributed communication can be realized through the domain ID mechanism (ROS_DOMAIN_ID) of DDS. Before starting the node, you can set the value of the domain ID. If the domain ID of different nodes is the same, they can discover and communicate freely. Otherwise, if the domain ID value is different, it cannot be realized. By default, all nodes are started with a domain ID of 0, in other words, as long as you ensure that you are on the same network, you do not need to do any configuration, different nodes on different ROS2 devices can achieve distributed communication.

The application scenarios of distributed communication are more extensive, such as unmanned vehicle formation, drone formation, remote control, etc. The interaction of these data all rely on distributed communication.

4.2. Implementation

4.2.1. Default implementation

You only need to put the host and slave [can have multiple] in the same network, you have achieved distributed communication. For example, the host and slave are connected to the same WiFi or the same router.

In Windows, if the network of the VM is set to Bridge mode, the VM and the host are on the same network.

Test:

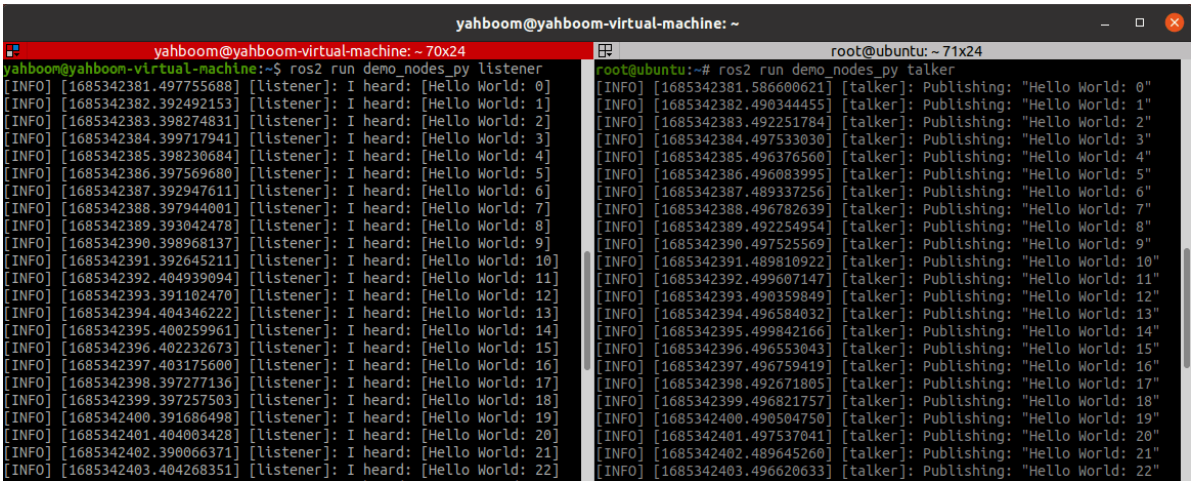
1. host end [car] execution:

```
ros2 run demo_nodes_py talker
```

2. At the same time, execute from the machine [virtual machine] :

```
ros2 run demo_nodes_py listener
```

If the following information is displayed: A topic published on the host can be subscribed to from the host in time, multi-host communication has been implemented



4.2.2. Distributed network grouping

If you are in a network with other robots in use, you can also set up a group for your robots in order not to be disturbed by other robots.

ROS2 provides a DOMAIN mechanism, just like grouping, computers in the same DOMAIN can communicate. We can add such a configuration in the ~/.bashrc of the host [car] and the slave [virtual machine] to assign the two to a group: :

```
export ROS_DOMAIN_ID=<your_domain_id>
```

If the ID assigned by the host [car] and the slave [virtual machine] is different, the two cannot communicate and achieve the purpose of grouping.

Note: The factory image of RDK X3 Robot car has been configured with the PC VM for multi-machine communication, and ROS_DOMAIN_ID is 66. You do not need to perform the following operations, and can directly test by referring to 4.2.1.

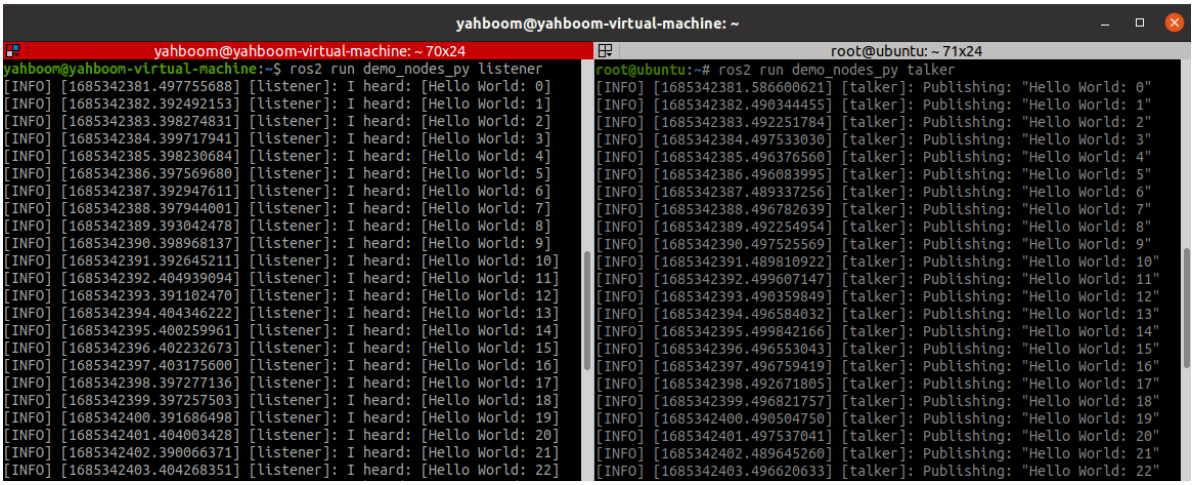
1. Execution of [car] on the host side:

```
echo "export ROS_DOMAIN_ID=66" >> ~/.bashrc  # 不一定要用66，符合ROS_DOMAIN_ID的规则即可                    # Doesn't have to use 66, just follow the rules for ROS_DOMAIN_ID
source ~/.bashrc
ros2 run demo_nodes_py talker
```

2. At the same time from the machine side [virtual machine] execution:

```
echo "export ROS_DOMAIN_ID=66" >> ~/.bashrc  # 这里和主机端的值保持一致
# This is the same as the host value
source ~/.bashrc
ros2 run demo_nodes_py listener
```

If the following information is displayed: The topic published on the host can be subscribed to from the host in time, it indicates that grouped multi-host communication has been implemented



> Note: Setting the value of ROS_DOMAIN_ID is not arbitrary, and there are certain constraints:
>
> 1. The recommended value for ROS_DOMAIN_ID is between [0,101], containing 0 and 101;
> 2. The total number of nodes in each domain ID is limited and must be less than or equal to 120.
> 3. If the domain ID is 101, the total number of nodes in the domain must be less than or equal to 54.

4.3 Calculation rules of DDS domain ID values

The rules for calculating domain ID values are as follows:

1. DDS is based on TCP/IP or UDP/IP network communication protocol, and the port number needs to be specified during network communication. The port number is represented by an unsigned integer of 2 bytes, and its value ranges from 0 to 65535.
2. The allocation of port numbers also has its rules, not can be used arbitrarily, according to the DDS protocol to 7400 as the starting port, you can also use the port is [7400,65535], and it is known that according to the DDS protocol by default, each domain ID occupies 250 ports, then the number of domain ids is: (65,535-7400)/250 = 232(each), the corresponding value range is [0,231];
3. The operating system also sets some reserved ports. When using ports in DDS, you need to avoid these reserved ports to avoid conflicts in use. Different operating systems have different reserved ports. The final result is that under Linux, available domain ids are [0,101] and [215-231]. The available domain ID on Windows and Mac is [0,166]. To be compatible with multiple platforms, it is recommended that the domain ID be in the range of [0,101].
4. Each domain ID occupies 250 ports by default, and each ROS2 node needs to occupy two ports. In addition, according to the DDS protocol, the first and second ports in the port segment of each domain ID are Discovery Multicast ports and User Multicast ports. Starting from the 11th and 12th ports, the Discovery Unicast port and User Unicast port of the first node in the domain, and the ports occupied by subsequent nodes are extended successively, then the maximum number of nodes in a domain ID is: (250-10)/2 = 120(number);
5. In special cases, if the domain ID is 101, the latter half of the domain port is reserved for the operating system, and the maximum number of nodes is 54.

The above calculation rules can be understood.