13、ROS2分布式通讯

1、概念

多机通讯即分布式通信是指可以通过网络在不同主机之间实现数据交互的一种通信策略。

ROS2本身是一个分布式通信框架，可以很方便的实现不同设备之间的通信，ROS2所基于的中间件是DDS，当处于同一网络中时，通过DDS的域ID机制(ROS_DOMAIN_ID)可以实现分布式通信，大致流程是：在启动节点之前，可以设置域ID的值，不同节点如果域ID相同，那么可以自由发现并通信，反之，如果域ID值不同，则不能实现。默认情况下，所有节点启动时所使用的域ID为0，换言之，只要保证在同一网络，你不需要做任何配置，不同ROS2设备上的不同节点即可实现分布式通信。

分布式通信的应用场景是较为广泛的，无人车编队、无人机编队、远程控制等等， 这些数据的交互都依赖于分布式通信。

2、实现

2.1、默认实现

只需要将主机和从机【可以有多个】处于同一个网络中，就已经实现了分布式通讯。比如主机和从机连接同一个WiFi或者同一个路由器。

Windows中虚拟机设置网络为【桥接模式】就和主机处于同一个网络了。

测试：

- 这里假设我们有两台主机A和B，可以是任何形式能连接网络的主机例如：虚拟机、树莓派、jetson、x86/ARM主机、卡片主板，只需要安装相同版本的ros2环境即可

1、A主机执行：

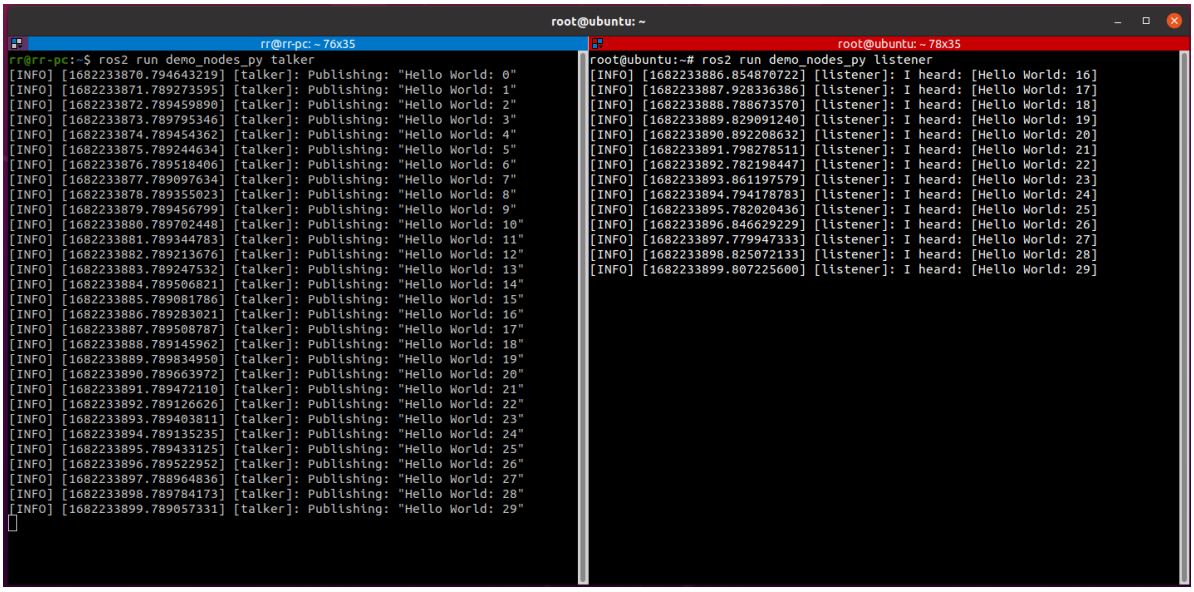
这里演示的是小车处于docker中，docker使用的网络模式是host模式，host模式简单来说就是和小车共用一个网络，所以跟在小车上执行没有区别。

```
ros2 run demo_nodes_py talker
```

2、B主机执行：

```
ros2 run demo_nodes_py listener
```

若显示如下：主机端发布的话题从机端能及时订阅到，表示已经实现了多机通讯



2.2、分布式网络分组

假设你现在所处的网络中还有其它的机器人在使用，为了不受其它机器人的干扰，你还可以给你的机器人设置一个分组。

ROS2提供了一个DOMAIN的机制，就类似分组一样，处于同一个DOMAIN中的计算机才能通信，我们可以在主机端【小车】和从机端【虚拟机】的.bashrc中加入这样一句配置，即可将两者分配到一个小组中：

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

如果主机端【小车】和从机端【虚拟机】分配的ID不同，则两者无法实现通信，达到分组的目的。

2.2.1 案例1

1、主机端【小车】执行：

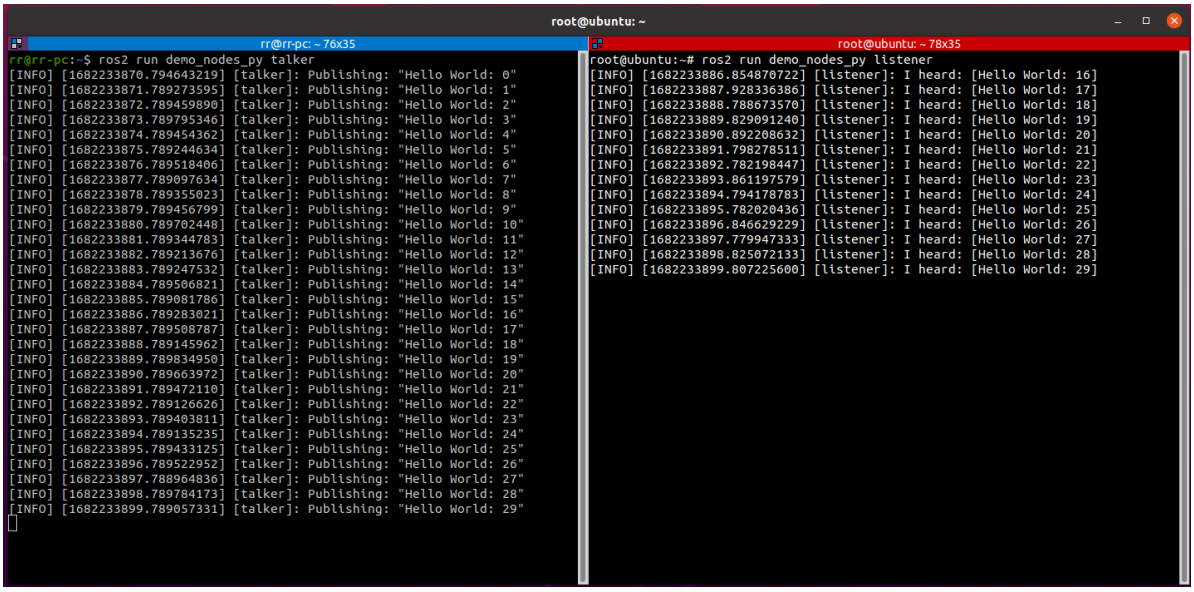
这里演示的是小车处于docker中，docker使用的网络模式是host模式，host模式简单来说就是和小车共用一个网络，所以跟在小车上执行没有区别。

```
echo "export ROS_DOMAIN_ID=6" >> ~/.bashrc  # 这里的6是ROS_DOMAIN_ID，不一定要用6，符合ROS_DOMAIN_ID的规则即可
source ~/.bashrc
ros2 run demo_nodes_py talker
```

2、同时从机端【虚拟机】执行：

```
echo "export ROS_DOMAIN_ID=6" >> ~/.bashrc  # 这里和主机端的值保持一致
source ~/.bashrc
ros2 run demo_nodes_py listener
```

若显示如下：主机端发布的话题从机端能及时订阅到，表示已经实现了分组的多机通讯



2.2.2 案例2

通过分布式通信控制小海龟运动

- 主机A运行指令

```
ros2 run turtlesim turtlesim_node
```

- 主机B运行指令

```
ros2 run turtlesim turtle_teleop_key
```

3、注意

在设置ROS_DOMAIN_ID的值时并不是随意的，也是有一定约束的：

1. 建议ROS_DOMAIN_ID的取值在[0,101] 之间，包含0和101；
2. 每个域ID内的节点总数是有限制的，需要小于等于120个；
3. 如果域ID为101，那么该域的节点总数需要小于等于54个。

4、DDS 域 ID 值的计算规则(进阶知识)

域ID值的相关计算规则如下：

1. DDS是基于TCP/IP或UDP/IP网络通信协议的，网络通信时需要指定端口号，端口号由2个字节的无符号整数表示，其取值范围在[0,65535]之间；
2. 端口号的分配也是有其规则的，并非可以任意使用的，根据DDS协议规定以7400作为起始端口，也即可用端口为[7400,65535]，又已知按照DDS协议默认情况下，每个域ID占用250个端口，那么域ID的个数为：(65535-7400)/250 = 232(个)，对应的其取值范围为[0,231]；
3. 操作系统还会设置一些预留端口，在DDS中使用端口时，还需要避开这些预留端口，以免使用中产生冲突，不同的操作系统预留端口又有所差异，其最终结果是，在Linux下，可用的域ID为[0,101]与[215-231]，在Windows和Mac中可用的域ID为[0,166]，综上，为了兼容多平台，建议域ID在[0,101] 范围内取值。

4. 每个域ID默认占用250个端口，且每个ROS2节点需要占用两个端口，另外，按照DDS协议每个域ID的端口段内，第1、2个端口是Discovery Multicast端口与User Multicast端口，从第11、12个端口开始是域内第一个节点的Discovery Unicast端口与User Unicast，后续节点所占用端口依次顺延，那么一个域ID中的最大节点个数为：(250-10)/2 = 120(个)；
5. 特殊情况：域ID值为101时，其后半段端口属于操作系统的预留端口，其节点最大个数为54个。

上述计算规则了解即可。