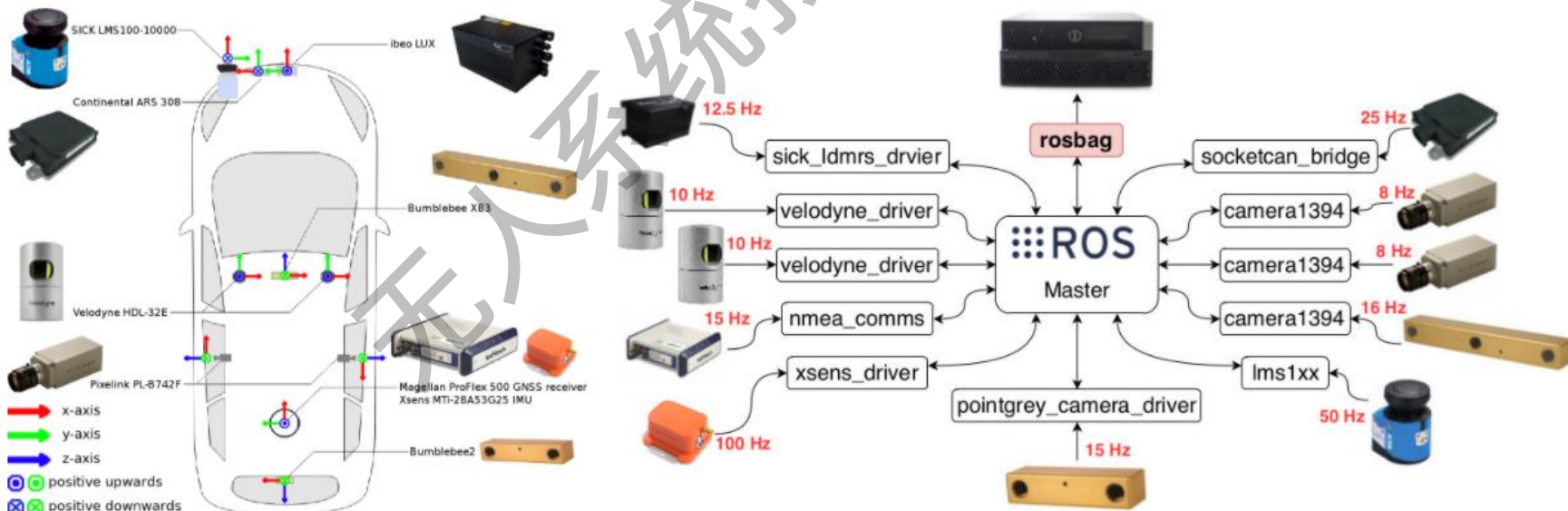
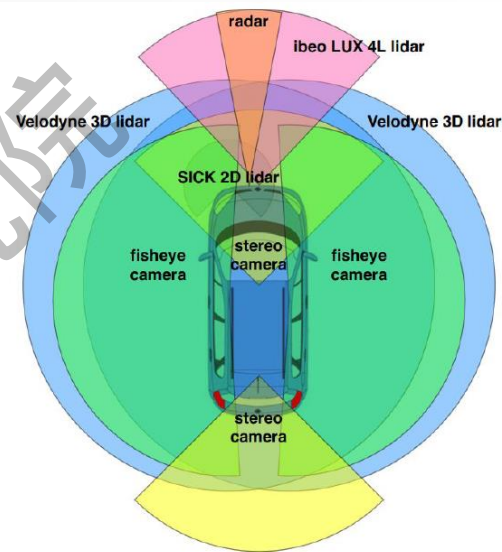
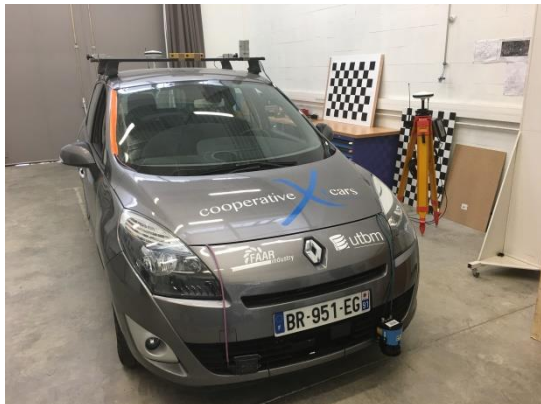




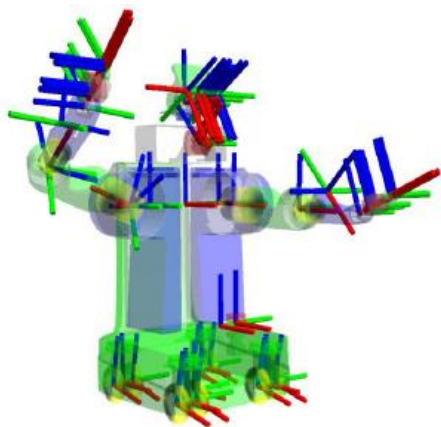
# 机器人操作系统 (ROS) 基础

- ◆ ROS 发展背景
- ◆ ROS 核心概念

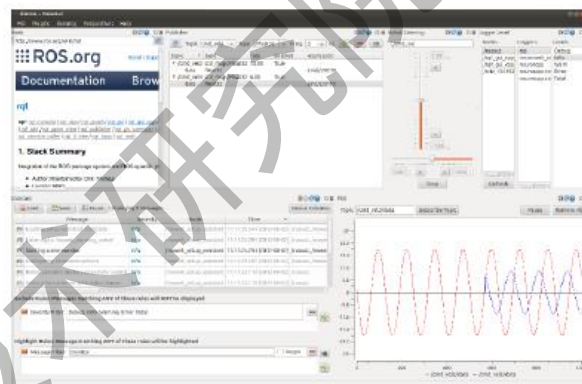




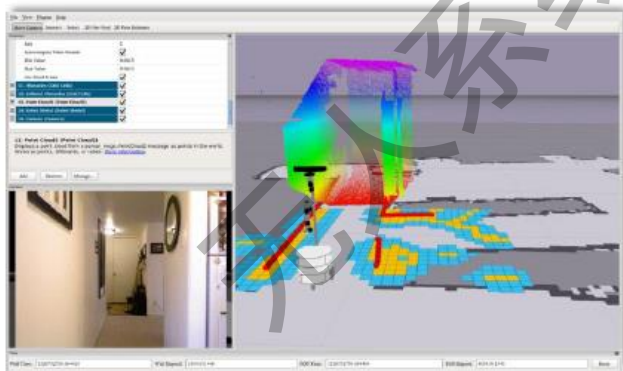
丰富的调试、仿真工具



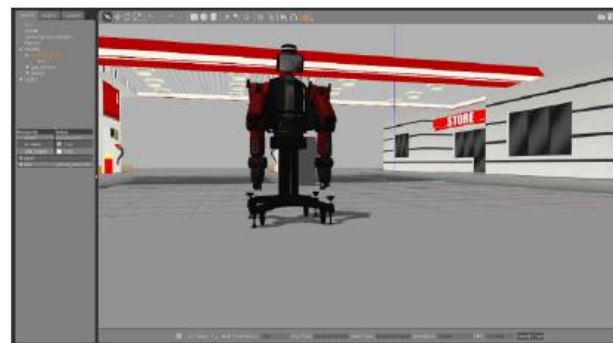
TF坐标变换



QT工具箱



Rviz



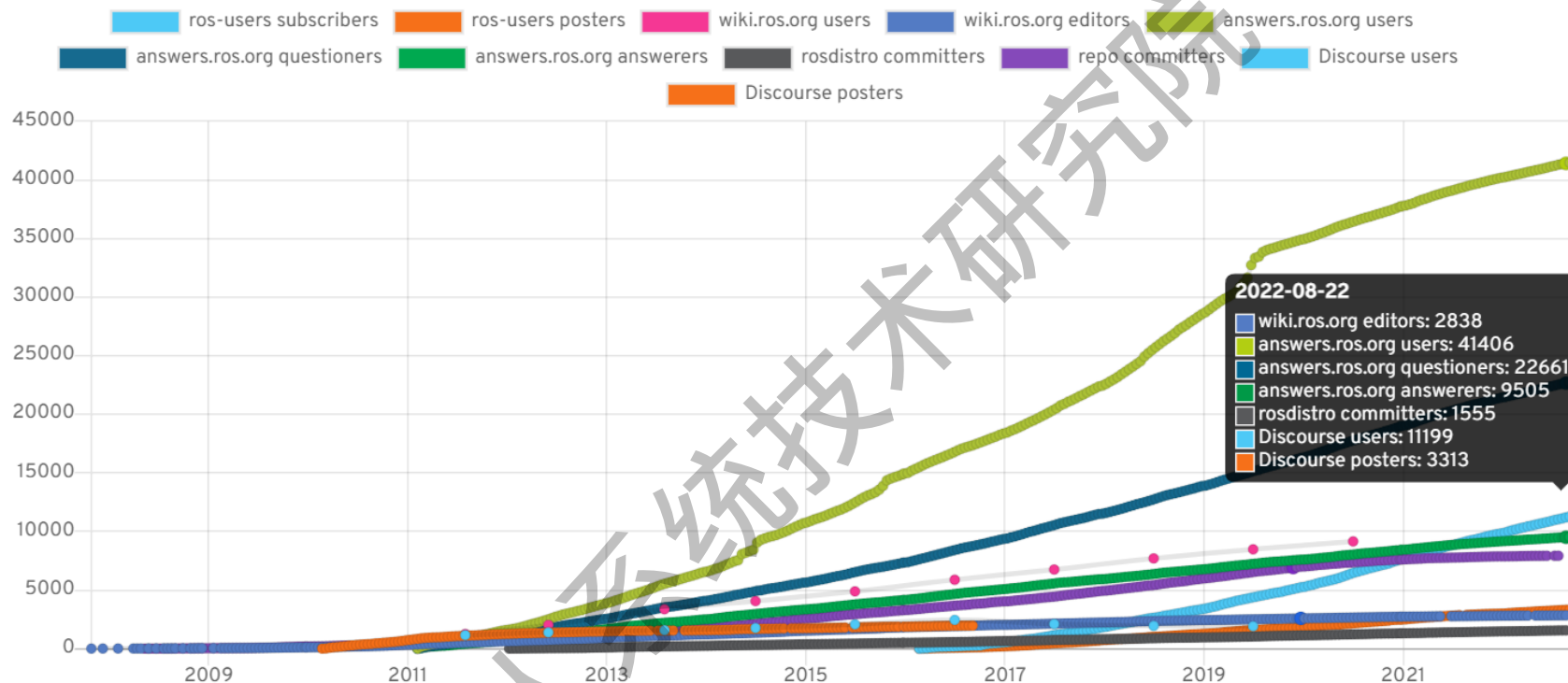
Gazebo



ROS版本	首选Ubuntu版本
Noetic	Ubuntu 20.04
Melodic	Ubuntu 18.04
Lunar	Ubuntu 17.04
Kinetic	Ubuntu 16.04
Jade	Ubuntu 15.04
Indigo	Ubuntu 14.04



Number of ROS Users



A collection of different metrics for measuring the number of users in the ROS community.

Overall	2014	2015	2016	2017	2018	2019	2020	2021	2022
1 United States us	1 United States us	United States us	United States us	United States us	United States us	United States us	United States us	United States us	United States us
2 China CN	2 Germany DE	Germany DE	Germany DE	China CN	China CN	China CN	Germany DE	Spain ES	Spain ES
3 Germany DE	3 Japan JP	Japan JP	China CN	Germany DE	Germany DE	Germany DE	Spain ES	China CN	China CN

Top ROS-using countries based on packages.ros.org downloads.

Source: <https://metrics.ros.org/>

智能无人系统综合设计



IEEE RA-L



IEEE ROBOTICS & AUTOMATION  
**LETTERS**  
A PUBLICATION OF THE IEEE ROBOTICS AND AUTOMATION SOCIETY

IEEE-RAM



IEEE T-RO



IEEE TRANSACTIONS ON  
**ROBOTICS**  
A PUBLICATION OF THE IEEE ROBOTICS AND AUTOMATION SOCIETY

IEEE 2021  
**ICRA**  
May 30 to June 5, 2021  
Xi'an, China

Paper submission due: October 31, 2020



2020 IEEE/RSJ

International Conference on  
Intelligent Robots and Systems (IROS)

October 25-29, 2020 Las Vegas, NV, USA



- Mobile manipulator : PR2
- Flight control system : pixhawk
- Multiple sensor: hokuyo sick pointgrey
- 更多: <http://wiki.ros.org/Robots>



[Fraunhofer IPA Care-O-bot](#)



[Videre Erratic](#)



[TurtleBot](#)



[Aldebaran Nao](#)



[Lego NXT](#)



[Shadow Hand](#)



[Willow Garage PR2](#)



[iRobot Roomba](#)



[Robotnik Guardian](#)



[Merlin miabotPro](#)



[AscTec Quadrotor](#)



[CoroWare Corobot](#)



[Clearpath Robotics Husky](#)



[Clearpath Robotics Kingfisher](#)



[Festo Didactic Robotino](#)



ROS 安装步骤: <http://wiki.ros.org/cn/melodic/Installation/Ubuntu>



The screenshot shows the ROS.org website with the following content:

- ROS.org** logo and navigation links: About | Support | Discussion Forum | Service Status | Q&A answers.ros.org
- Search:** [Input field] [Submit button]
- Documentations** | **Browse Software** | **News** | **Download**
- cn | melodic | Installation | Ubuntu**
- # 在 Ubuntu 中安装 ROS Melodic
- 我们编译几个 Ubuntu 平台下的 Debian 软件包, 直接安装编译好的软件包比从源码编译安装更加高效, 这也是我们在 Ubuntu 上的首选安装方式。注意, 还有从 Ubuntu 上游提供的包。请参见 [Upstream Packages \(英文网页\)](#) 以了解其中的区别。
- 我们为下列 Ubuntu 发行版编译了安装包。
- | Distro | amd64 | arm64 | armhf |
|--------|-------|-------|-------|
| Artful | X     |       |       |
| Bionic | X     | X     | X     |
- 如果你需要从源码编译安装(不推荐), 请参阅从源代码安装(下载和编译)。
- ★ 如果你使用了这些软件包, 请支持 OSRF.**

These packages are built and hosted on infrastructure maintained and paid for by the [Open Source Robotics Foundation](#), a 501(c)(3) non-profit organization. If OSRF were to receive one penny for each downloaded package for just two months, we could cover our annual costs to manage, update, and host all of our online services. Please consider [donating to OSRF today](#).
- Sommaire**
  - 1. 在 Ubuntu 中安装 ROS Melodic
    - 1. 安装
      - 1. 配置 Ubuntu 软件仓库
      - 2. 添加 sources.list
      - 3. 添加公钥
- Wiki**
  - [Distributions](#)
  - [ROS/Installation](#)
  - [ROS/Tutorials](#)
  - [RecentChanges](#)
  - Ubuntu**
- Page**
  - Page immuable
  - [Informations](#)
  - [Pièces jointes](#)
- Autres actions :**
  - Code source [v]
  - [Exécuter](#)
- Utilisateur**
  - [Connexion](#)



## 1.2 ROS 核心概念

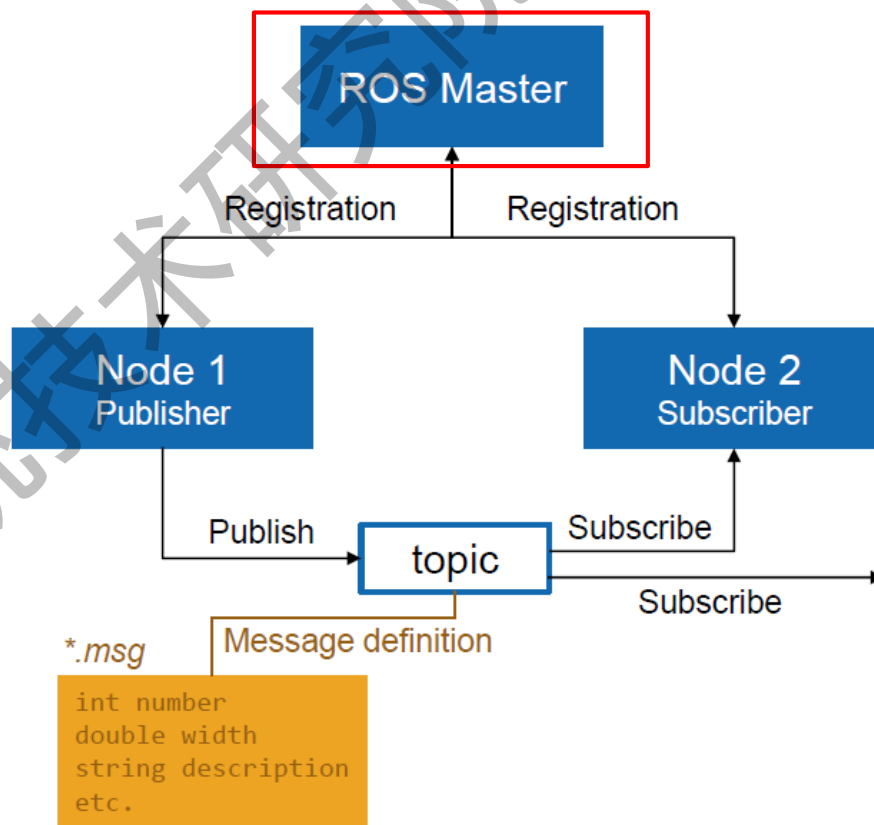


节点管理器 (ROS Master) :

- 为节点提供命名和注册服务;
- 跟踪和记录话题/服务通信, 辅助节点相互查找、建立连接;
- 提供参数服务器, 节点使用此服务器存储和检索运行时的参数。

运行:

> Roscore



## 1.2 ROS 核心概念



节点 (Node) :

- 执行具体任务的进程、独立运行的可执行文件;
- 不同节点可使用不同的编程语言, 可分布式运行在不同的主机;
- 节点在系统中的名称必须是唯一的, 由包 (packages) 组织。

运行:

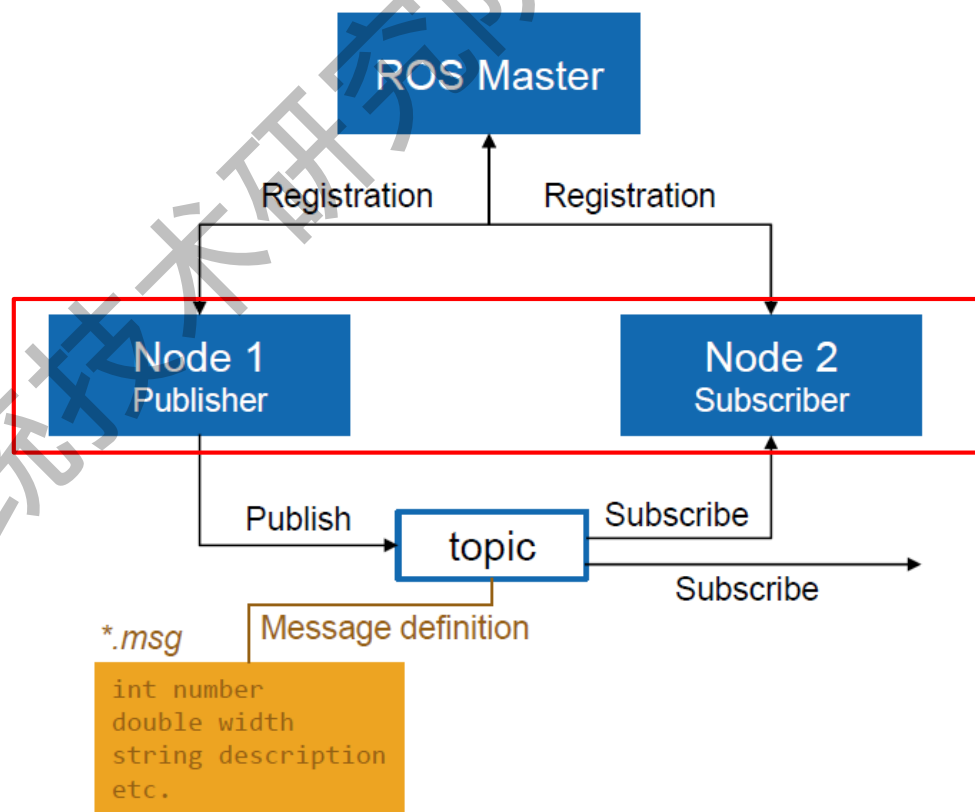
> `roslaunch package_name node_name`

查看激活的节点:

> `roslaunch list`

查看节点信息:

> `roslaunch info node_name`



## 1.2 ROS 核心概念



话题 (Topic) :

- 节点间用来传输数据的重要总线;
- 使用发布/订阅模型, 数据由发布者传输到订阅者, 同一个话题的订阅者或发布者可以不唯一。

查看激活的话题:

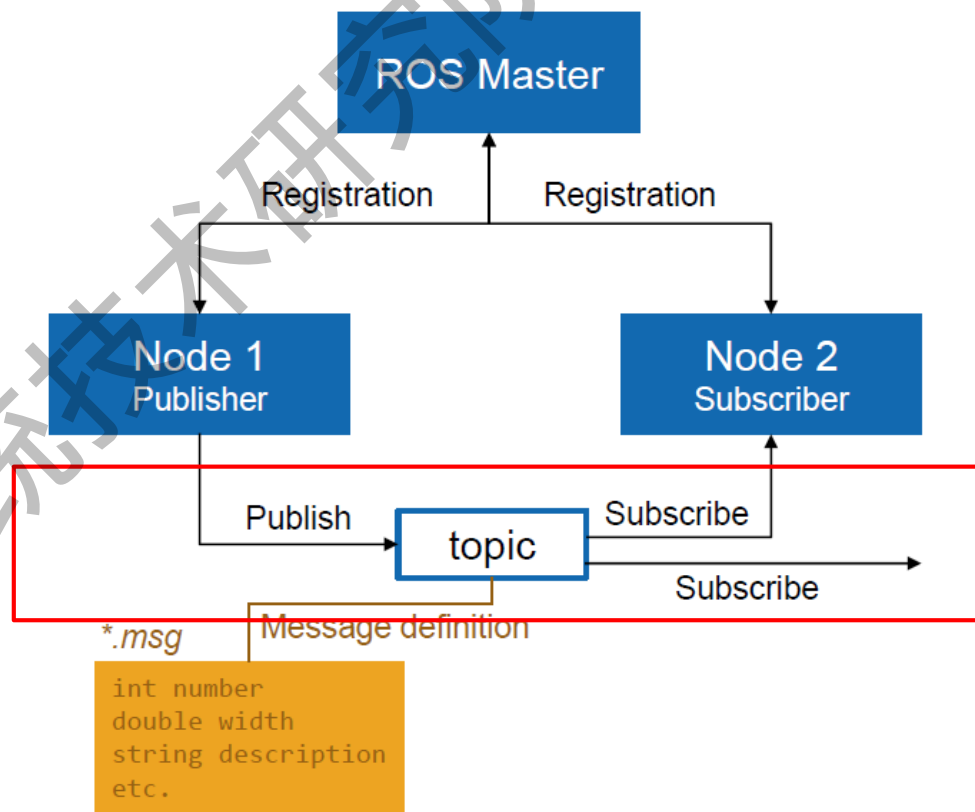
> **rostopic list**

查看话题输出:

> **rostopic echo /topic**

查看话题信息:

> **rostopic info /topic**



## 1.2 ROS 核心概念



消息 (Message) :

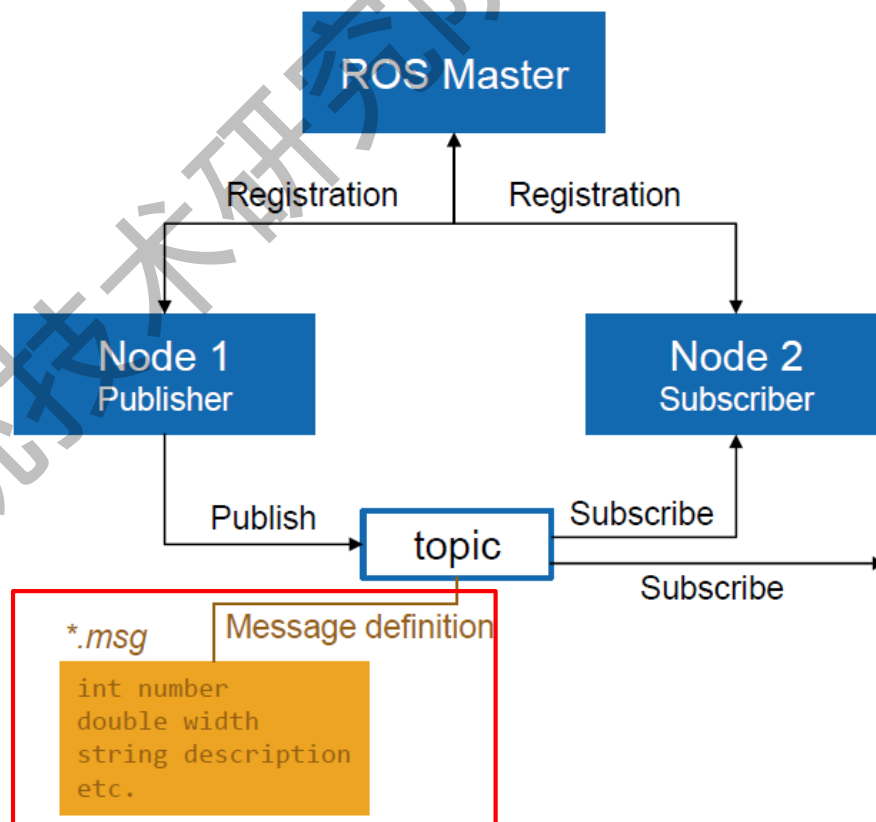
- 具有一定的类型和数据结构, 包括 ROS 提供的标准类型和用户自定义类型;
- 使用编程语言无关的 .msg 文件定义, 编译过程中生成对应的代码文件。

查看话题类型:

> **rostopic type /topic**

向话题输出消息:

> **rostopic pub /topic type data**





消息 (Message) :

geometry\_msgs/Point.msg

```
float64 x  
float64 y  
float64 z
```

sensor\_msgs/Image.msg

```
std_msgs/Header header  
  uint32 seq  
  time stamp  
  string frame_id  
uint32 height  
uint32 width  
string encoding  
uint8 is_bigendian  
uint32 step  
uint8[] data
```

geometry\_msgs/PoseStamped.msg

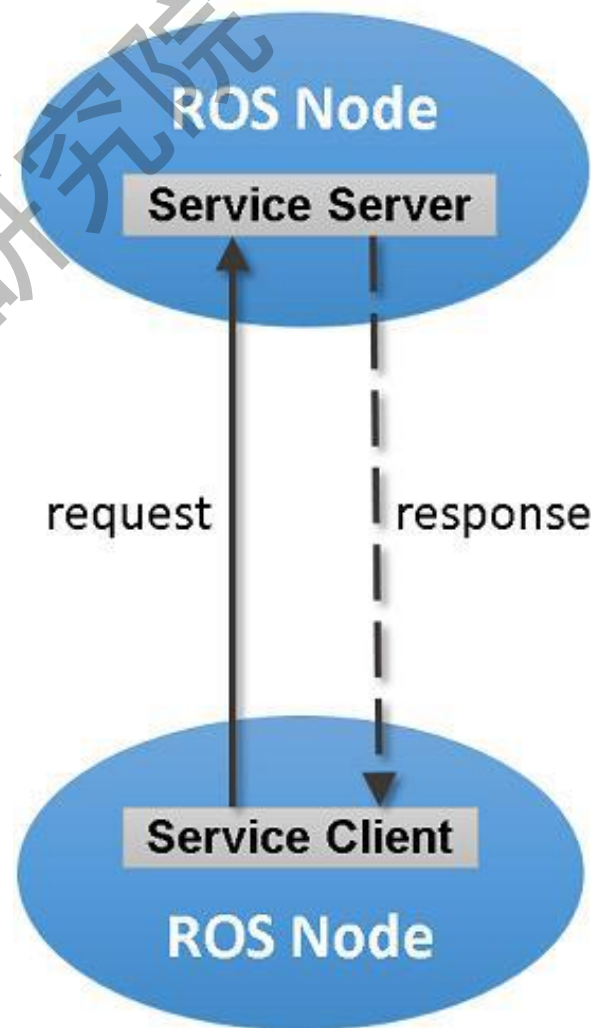
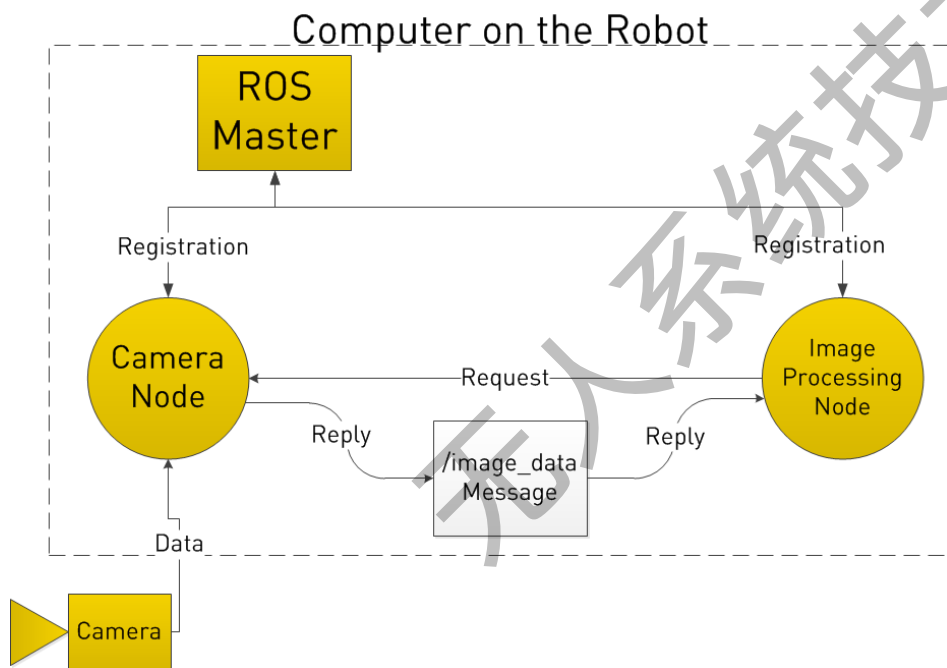
```
std_msgs/Header header  
uint32 seq  
time stamp  
string frame_id  
geometry_msgs/Pose pose  
→ geometry_msgs/Point position  
  float64 x  
  float64 y  
  float64 z  
geometry_msgs/Quaternion orientation  
  float64 x  
  float64 y  
  float64 z  
  float64 w
```

## 1.2 ROS 核心概念



服务 (Service) :

使用客户端/服务器 (Client/Service) 模型，客户端发送请求数据，服务器完成处理后返回应答数据。



智能无人系统综合设计



## 话题和服务的异同

	话题	服务
同步性	异步	同步
通信模型	发布/订阅	服务器/客户端
底层协议	ROSTCP/ROSUDP	ROSTCP/ROSUDP
反馈机制	无	有
缓冲区	有	无
实时性	弱	强
节点关系	多对多	一对多 (一个server)
适用场景	数据传输	逻辑处理

## 1.2 ROS 核心概念



运行:

> roscore



```
student@ubuntu:~/catkin_ws$ roscore
... logging to /home/student/.ros/log/6c1852aa-e961-11e6-8543-000c297bd368/ros
launch-ubuntu-6696.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://ubuntu:34089/
ros_comm version 1.11.20

SUMMARY
=====
PARAMETERS
* /rostdistro: indigo
* /rosversion: 1.11.20

NODES

auto-starting new master
process[master]: started with pid [6708]
ROS_MASTER_URI=http://ubuntu:11311/

setting /run_id to 6c1852aa-e961-11e6-8543-000c297bd368
process[rosout-1]: started with pid [6721]
started core service [/rosout]
```





## 1.2 ROS 核心概念



运行:

> roscore

运行一个发布者:

> rosruncatkin\_ws\$ roscpp\_tutorials\_talker

```
student@ubuntu:~/catkin_ws$ rosruncatkin_ws$ roscpp_tutorials_talker
[ INFO] [1486051708.424661519]: hello world 0
[ INFO] [1486051708.525227845]: hello world 1
[ INFO] [1486051708.624747612]: hello world 2
[ INFO] [1486051708.724826782]: hello world 3
[ INFO] [1486051708.825928577]: hello world 4
[ INFO] [1486051708.925379775]: hello world 5
[ INFO] [1486051709.024971132]: hello world 6
[ INFO] [1486051709.125450960]: hello world 7
[ INFO] [1486051709.225272747]: hello world 8
[ INFO] [1486051709.325389210]: hello world 9
```



## 1.2 ROS 核心概念



运行:

> **roscore**

运行一个发布者:

> **roslaunch roscpp\_tutorials talker**

查看激活的节点:

> **roslaunch list**

查看节点信息:

> **roslaunch info /talker**

查看话题信息:

> **rostopic info /chatter**

```
student@ubuntu:~/catkin_ws$ roslaunch list
/roslaunch
/talker
```

```
student@ubuntu:~/catkin_ws$ roslaunch info /talker
```

```
-----
--
Node [/talker]
Publications:
* /chatter [std_msgs/String]
* /roslaunch [roslaunch_msgs/Log]

Subscriptions: None

Services:
* /talker/get_loggers
* /talker/set_logger_level
```

```
student@ubuntu:~/catkin_ws$ rostopic info /chatter
Type: std_msgs/String

Publishers:
* /talker (http://ubuntu:39173/)

Subscribers: None
```



## 1.2 ROS 核心概念



运行:

> **roscore**

运行一个发布者:

> **roslaunch roscpp\_tutorials talker**

查看激活的节点:

> **rostopic list**

查看节点信息:

> **rostopic info /talker**

查看话题信息:

> **rostopic info /chatter**

查看话题类型:

> **rostopic type /chatter**

显示话题消息:

> **rostopic echo /chatter**

输出话题频率:

> **rostopic hz /chatter**

```
student@ubuntu:~/catkin_ws$ rostopic type /chatter
std_msgs/String
```

```
student@ubuntu:~/catkin_ws$ rostopic echo /chatter
data: hello world 11874
---
data: hello world 11875
---
data: hello world 11876
```

```
student@ubuntu:~/catkin_ws$ rostopic hz /chatter
subscribed to [/chatter]
average rate: 9.991
  min: 0.099s max: 0.101s std dev: 0.00076s window: 10
average rate: 9.996
  min: 0.099s max: 0.101s std dev: 0.00069s window: 20
```



## 1.2 ROS 核心概念



运行:

> **roscore**

运行一个发布者:

> **roslaunch roscpp\_tutorials talker**

查看激活的节点:

> **rostopic list**

查看节点信息:

> **rostopic info /talker**

查看话题信息:

> **rostopic info /chatter**

查看话题类型:

> **rostopic type /chatter**

显示话题消息:

> **rostopic echo /chatter**

输出话题频率:

> **rostopic hz /chatter**

运行一个订阅者:

> **roslaunch roscpp\_tutorials listener**

```
student@ubuntu:~/catkin_ws$ roslaunch roscpp_tutorials listener
[ INFO] [1486053802.204104598]: I heard: [hello world 19548]
[ INFO] [1486053802.304538827]: I heard: [hello world 19549]
[ INFO] [1486053802.403853395]: I heard: [hello world 19550]
[ INFO] [1486053802.504438133]: I heard: [hello world 19551]
[ INFO] [1486053802.604297608]: I heard: [hello world 19552]
```





## 1.2 ROS 核心概念

运行:

```
> roscore
```

启动小海龟仿真器:

```
> rosrunc turtlesim turtlesim_node
```

启动小海龟控制器:

```
> rosrunc turtlesim turtle_teleop_key
```

小海龟仿真器界面



## 1.2 ROS 核心概念



运行:

> **roscore**

启动小海龟仿真器:

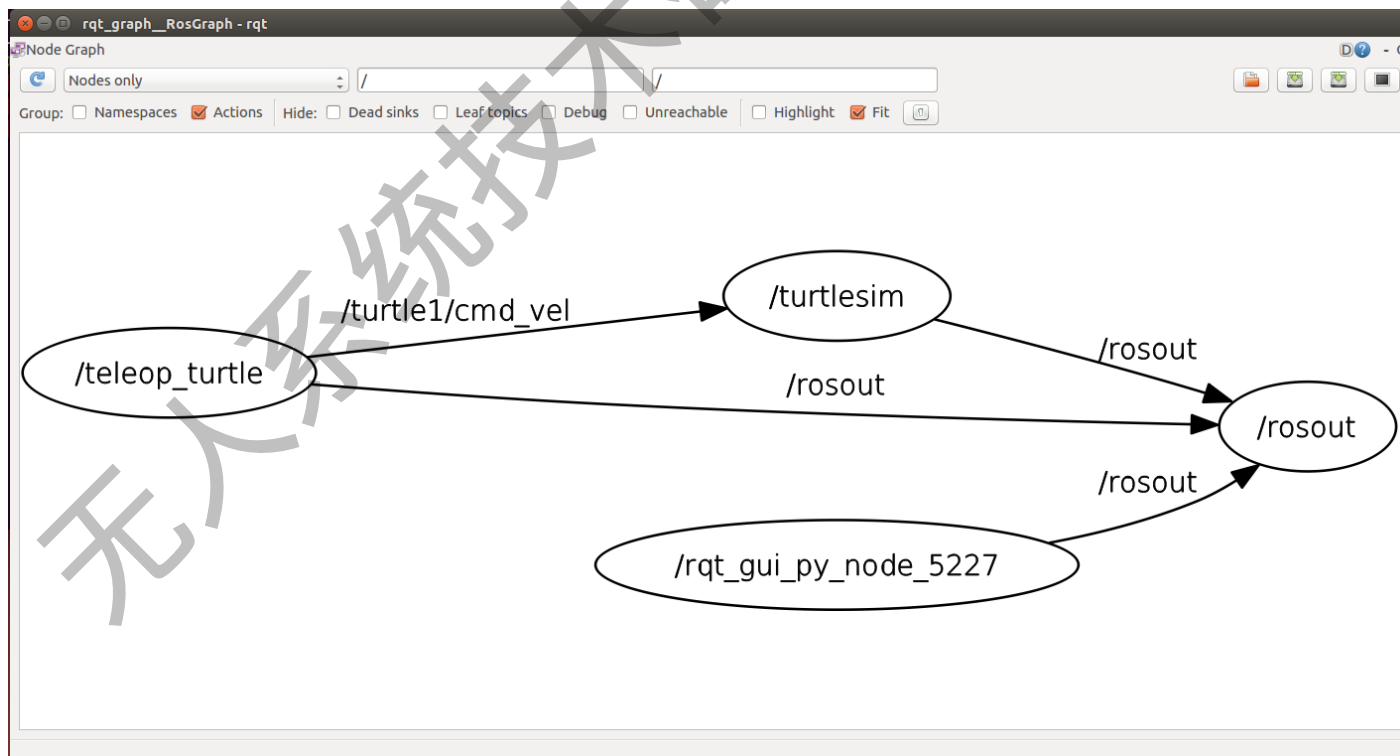
> **roslaunch turtlesim turtlesim\_node**

启动小海龟控制器:

> **roslaunch turtlesim turtle\_teleop\_key**

查看系统运行图:

> **rqt\_graph**



## 1.2 ROS 核心概念

运行:

```
> roscore
```

启动小海龟仿真器:

```
> rosrunc turtlesim turtlesim_node
```

启动小海龟控制器:

```
> rosrunc turtlesim turtle_teleop_key
```

查看系统运行图:

```
> rqt_graph
```

让小海龟连续转圈:

```
> rostopic pub /turtle1/cmd_vel -r 10 geometry_msgs/Twist '{linear: {x: 0.2, y: 0, z: 0},  
angular: {x: 0, y: 0, z: 0.5}}'
```



## 1.2 ROS 核心概念

运行:

```
> roscore
```

启动小海龟仿真器:

```
> rosrunc turtlesim turtlesim_node
```

启动小海龟控制器:

```
> rosrunc turtlesim turtle_teleop_key
```

查看系统运行图:

```
> rqt_graph
```

让小海龟连续转圈:

```
> rostopic pub /turtle1/cmd_vel -r 10 geometry_msgs/Twist '{linear: {x: 0.2, y: 0, z: 0},  
angular: {x: 0, y: 0, z: 0.5}}'
```

记录所有话题:

```
> rosbag record -a -O cmd_vel_record
```

回放记录的话题:

```
> rosbag play cmd_vel_record.bag
```





## 1.2 ROS 核心概念



运行:

```
> roscore
```

启动小海龟仿真器:

```
> rosrun turtlesim turtlesim_node
```

启动小海龟控制器:

```
> rosrun turtlesim turtle_teleop_key
```

查看系统运行图:

```
> rqt_graph
```

让小海龟连续转圈:

```
> rostopic pub /turtle1/cmd_vel -r 10 geometry_msgs/Twist '{linear: {x: 0.2, y: 0, z: 0},  
angular: {x: 0, y: 0, z: 0.5}}'
```

记录所有话题:

```
> rosbag record -a -O cmd_vel_record
```

回放记录的话题:

```
> rosbag play cmd_vel_record.bag
```

调一只新的海龟:

```
> rosservice call spawn 2 2 0.1 "turtle2"
```



## 1.2 ROS 核心概念



工作空间 (workspace) 是一个存放工程开发相关文件的文件夹：

- **src**: 代码空间 (Source Space)
- **build**: 编译空间 (Build Space)
- **devel**: 开发空间 (Development Space)

Work here    Don't touch    Don't touch



src



build



devel





课程结束，欢迎提问

THANK YOU FOR WATCHING