BUSINESS 2019

JULYED

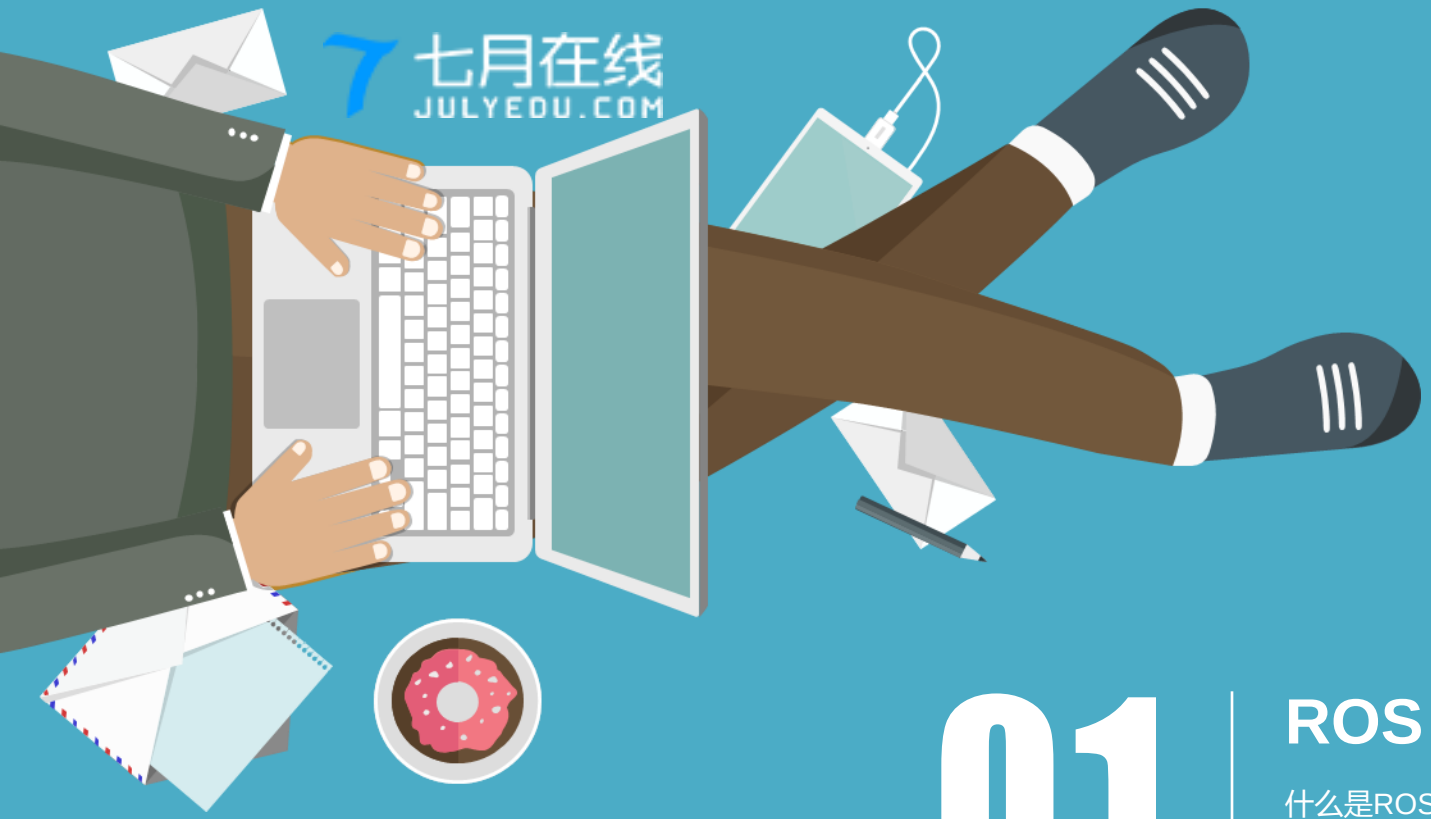
ROS概述

北京七月在线科技有限公司

张老师 无人驾驶高级算法工程师

https://www.julyedu.com/


七月在线
JULYEDU.COM



01 ROS 系统概述

什么是ROS

什么是ROS

- ROS: The Robot Operating System

- 2007年，起源于Stanford AI实验室，为了支持STAIR机器人而建立的交换庭(switchyard)项目。后又与Willow Garage公司的个人机器人项目（Personal Robots Program）之间合作。
- 2008年，主要由Willow Garage来进行推动。随着PR2那些不可思议的表现，ROS得到越来越多的关注。
- 2010年，Willow Garage正式发布ROS1.0
- 2013年，Open Source Robotics Foundation接手维护
- 2016年，正式发布ROS2.0

什么是ROS

本课程使用kinetic版本

为什么使用ROS

- 点对点设计
- 分布式设计
- 多语言
- 轻量级
- 免费且开源
- 社区完善

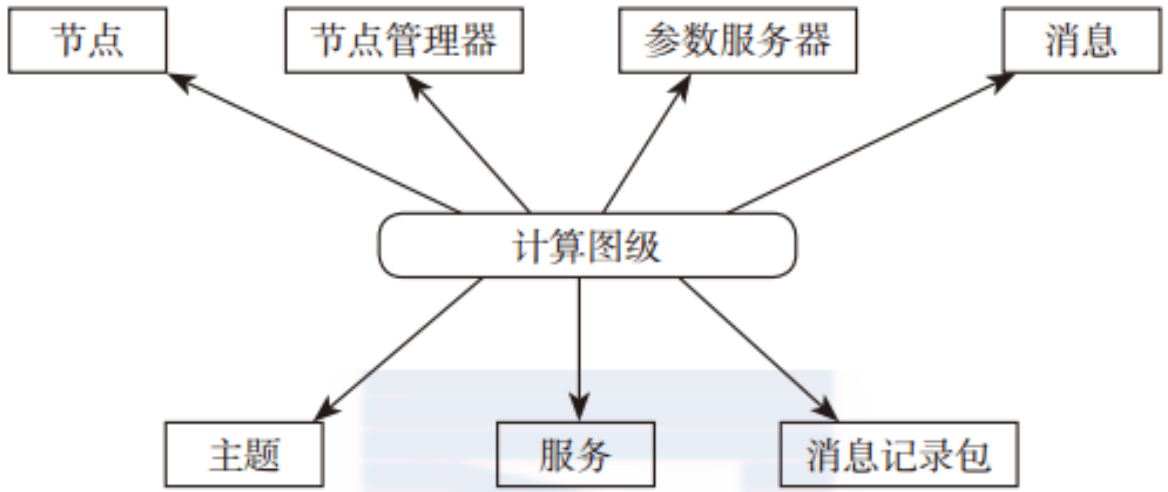
http://www.ros.org

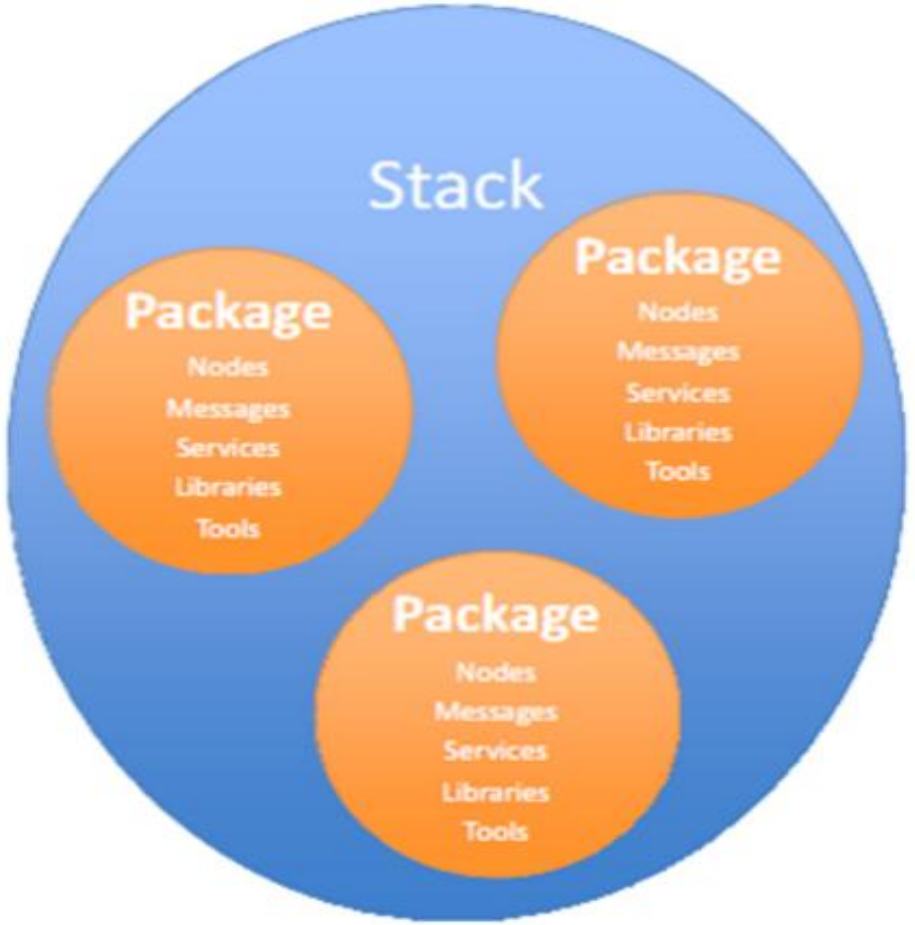
ROS不是

- ROS （Robot Operating System）
- 不是传统意义上的一种操作系统，而是一种系统软件框架，该框架使用了流行的面向服务（SOA）的软件技术，通过网络协议将节点间数据通信解耦。这样就能够轻松地集成不同语言不同功能的代码。

- ROS不是一种编程语言
- ROS不仅是一个函数库，除包含客户端（Client Libraries）外，还包含一个中心服务器（Central Server）、一系列命令行工具、图形化界面工具以及编译环境。
- ROS不是集成开发环境。

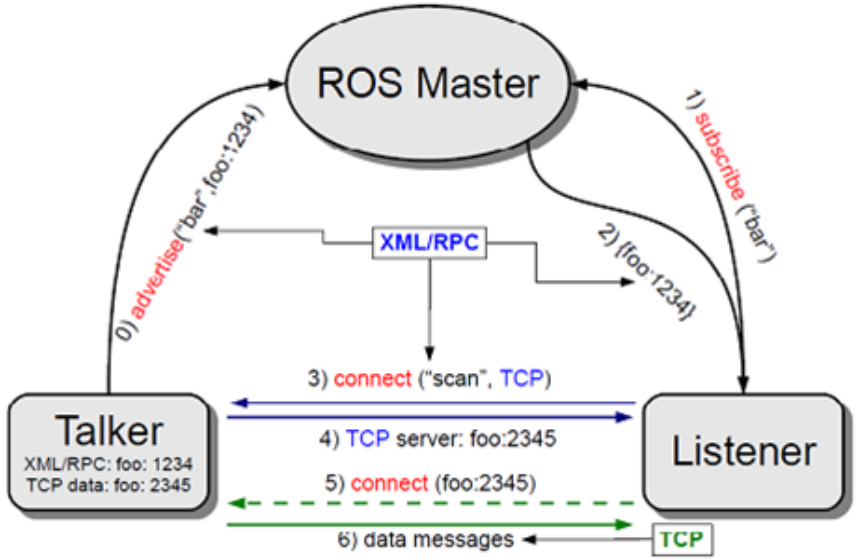
ROS


节点
节点管理器
参数服务器
消息
计算图级
主题
服务
消息记录包

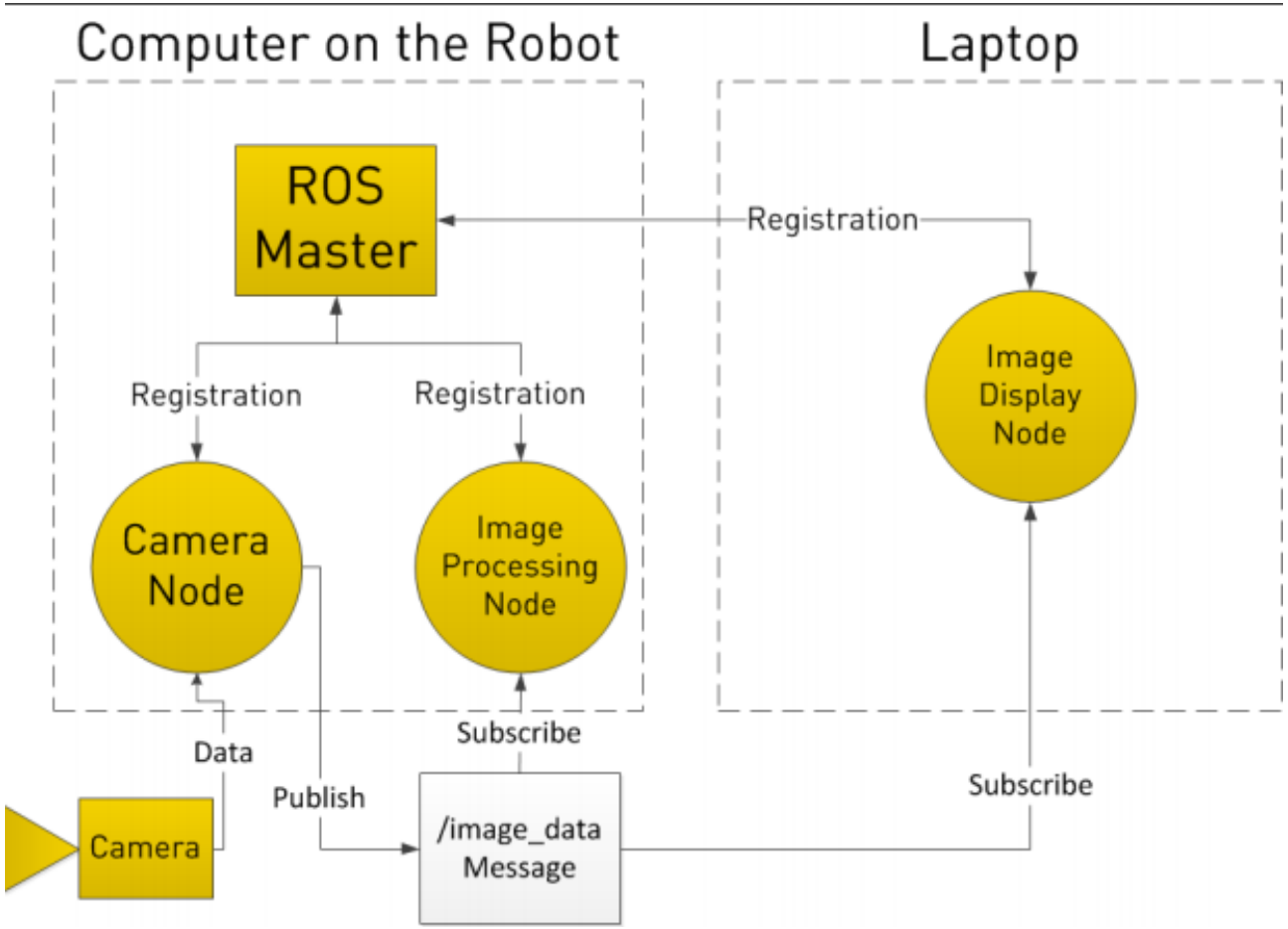

Stack
Package
Nodes
Messages
Services
Libraries
Tools
Package
Nodes
Messages
Services
Libraries
Tools
Package
Nodes
Messages
Services
Libraries
Tools

ROS

- 1. 消息以一种publish/subscribe的方式传递
- 2. 节点可以在给定的主题中发布/订阅消息
- 3. 一个节点可以订阅/发布多个不同的主题
- 4. 允许多个节点订阅/发布同一个主题
- 5. 订阅节点和发布节点并不知道相互之间的存在

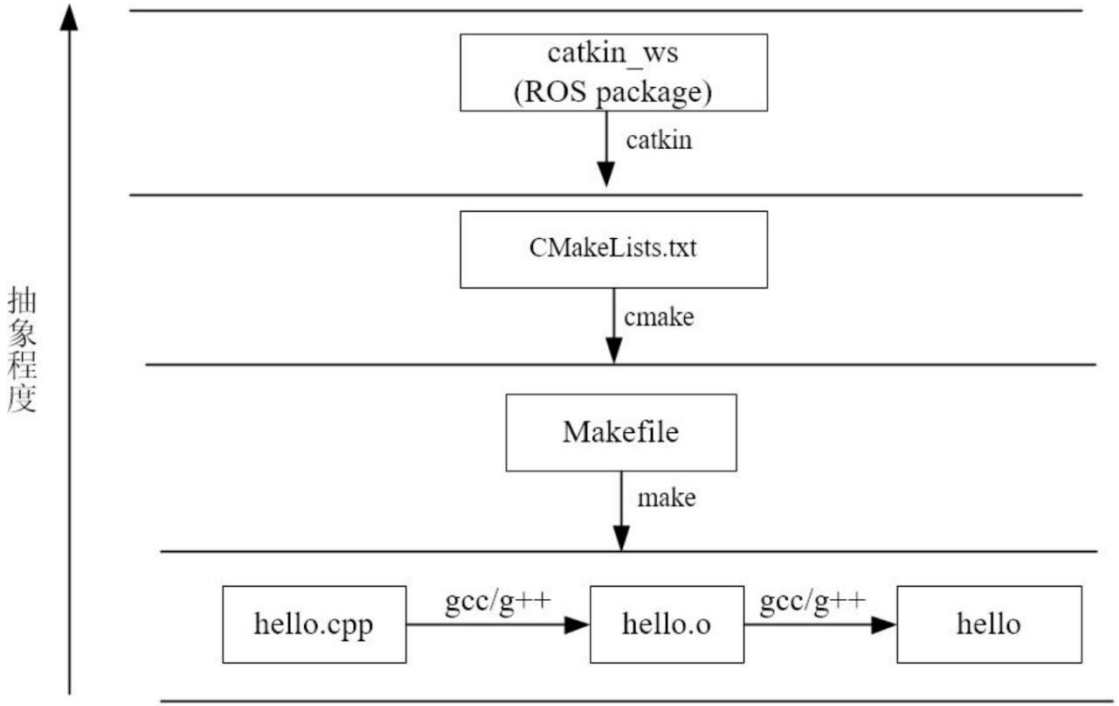


ROS


Computer on the Robot
Laptop
ROS Master
Registration
Registration
Registration
Image Display Node
Camera Node
Image Processing Node
Data
Subscribe
Publish
Subscribe
Camera
/image_data Message

Catkin

- ROS的编译系统
- ROS对CMake进行了扩展
- 适合ROS进行大型项目，多包，多节点的场景下进行批量编译



•操作更加简单
•一次配置，多次使用
•跨依赖项目编译

Packages & Catkin Workspaces

- Packages

  Package是ROS系统中最底层最基本的组织，里面存放各种文件:库、工具、可执行文件等.

- **Catkin workspaces**
- 包的顶层工作目录，一个catkin workspace 包含一个工程下面多个ros package

```
workspace_folder/        --WORKSPACE
 src/                   --SOURCE SPACE
    CMakeLists.txt/      --This is symlinked to catkin/cmake/toplevel.cmake
    package_1/
      CMakeLists.txt
      package.xml
    ...
    package_n/
      CMakeLists.txt
      package.xml
 build/
 devel/     --DEVEL SPACE (targets go here, parameterizable, but defaults to peer of Build Space)
    bin/
    etc/
    /include/
    lib/
    share/
    .catkin  --Marking the folder as a development space (the file contains a semicolon separated list of Source space paths)
                           #
    env.bash
    setup.bash
    setup.sh
```

Package.xml

- 每个包的描述文件,都需要放置在包的根目录下,对包的名字/版本/作者/维护者/依赖关系进行说明

```
<package>
  <name>july_test_package</name>
  <version>1.2.4</version>
  <description>
    This package provides foo capability.
  </description>
  <maintainer email="zy@july.com">s</maintainer>
  <license>BSD</license>

  <url>http://ros.org/wiki/foo_core</url>
  <author>zy</author>
  <depend>roscpp</depend>
  <depend>std_msgs</depend>
  <build_depend>message_generation</build_depend>
  <exec_depend>message_runtime</exec_depend>
  <exec_depend>rospy</exec_depend>
</package>
```

CmakeList.txt

- CmakeList.txt
  定义一个包的的编译指令
- cmake 不会找package.xml文件. 依据cmakelists.txt文件编译需要清晰指出头文件和库文件的指向.
- catkin_package(CATKIN_DEPENDS roscpp) 声明依赖本包同时需要的其他ros包
- find_package(catkin REQUIRED COMPONENTS ...)声明编译本包所需要的其他ros包
- add_executable 声明编译本包生成的可执行文件
- target_link_libraries 链接可执行文件和依赖库

ROS NODE

- 一个节点是ROS程序包中的一个可执行文件
- ROS节点可以使用ROS客户库与其他节点通信
- 节点可以发布或者接受一个话题。
- 节点可以提供或使用某项服务

- 常用命令
- rosnode list 查看当前注册到ros master 的所有节点
- Rosnode info 查看某个节点的具体信息

ROS TOPIC

- 节点之间是通过一个ROS **topic** 来互相通信的。
- 通过publisher 声明所发布 topic 名称
- 通过subscriber 声明所需要监听的topic名称

- 常用命令
- Topic list 查看当前注册到ros master 的所有topic 列表
- rostopic echo 把当前topic输出到控制台，方便调试和查看

ROS PARAM

- Rosparam命令允许我们在ROS 参数服务器上存储和复制数据。参数服务器可以存储整数、浮点数、布尔值、字典和列表值。
- 一个小型的KV库

常用命令

rosparam set	设置参数
rosparam get	获取参数
rosparam list	查看当前服务器上的参数

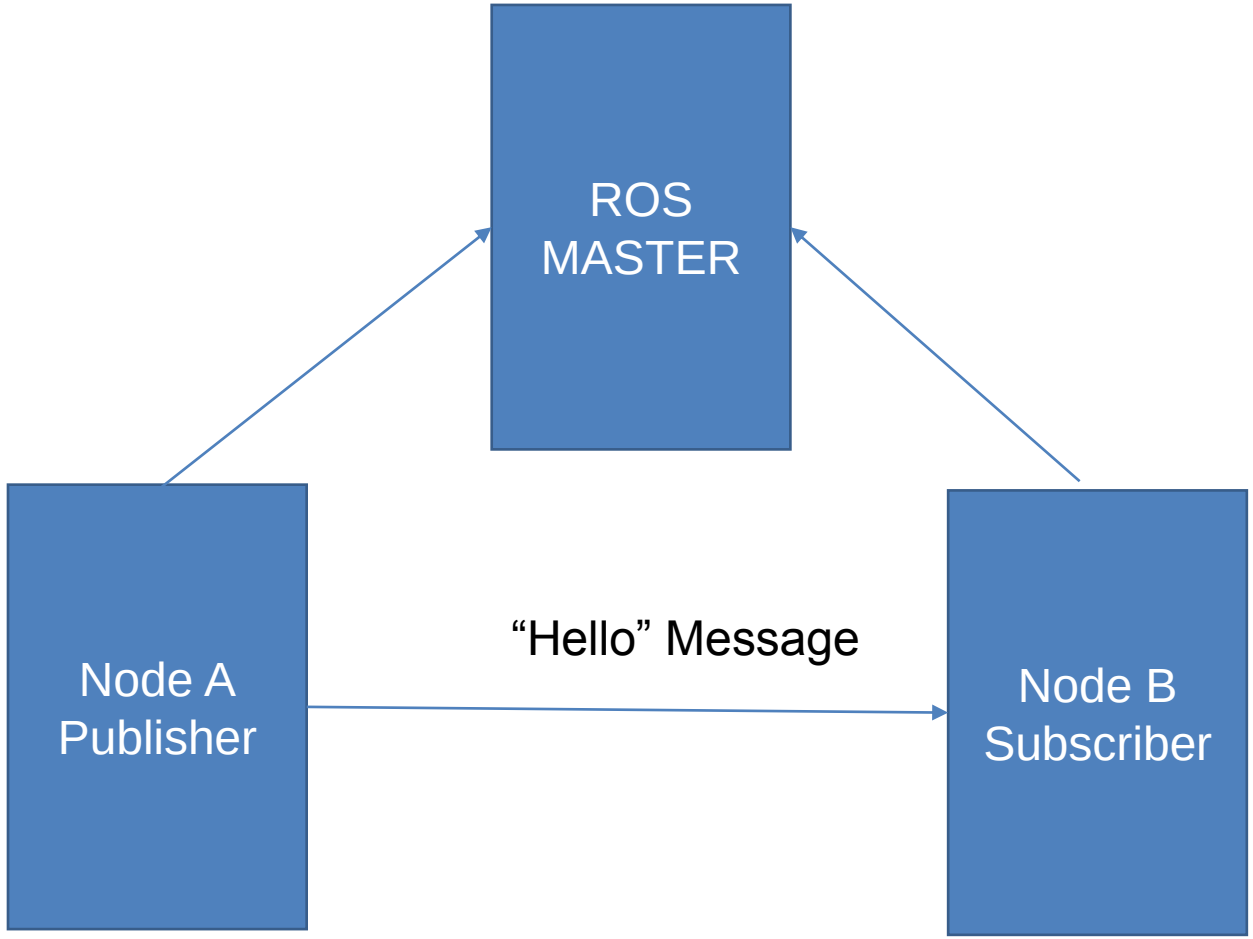
Listener & Talker

- 监听者-发布者 实践

Publisher
发布数据一方

Subscriber
订阅数据的一方

Message
数据类型



安装ROS-kinetic

- http://wiki.ros.org/kinetic/Installation/Ubuntu
- 安装注意，国内安装使用清华源
- sudo sh -c '. /etc/lsb-release && echo "deb http://mirrors.tuna.tsinghua.edu.cn/ros/ubuntu/ $DISTRIB_CODENAME main" > /etc/apt/sources.list.d/ros-latest.list'

Create a ROS Workspace

- mkdir -p ~/catkin_ws/src
- cd ~/catkin_ws
- catkin_make
- 添加devel下的setup.bash 到 bashrc中

创建一个rospackage

- catkin_create_pkg july_say std_msgs roscpp
- 在july_say/src下编写代码
- 重点代码

  ros::Publisher july_pub = n.advertise<std_msgs::String>(”july_topic“, 1000); //声明往july_topic发布std_msgs类型的message

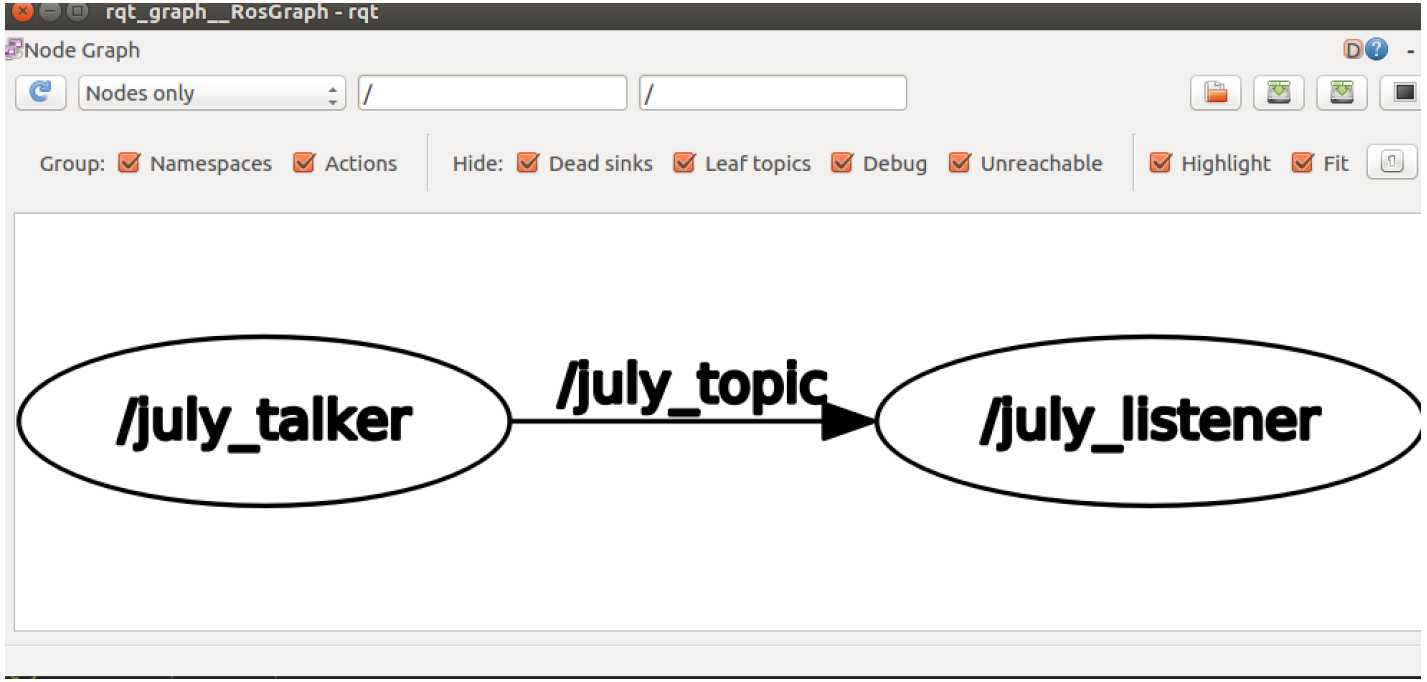
创建july_listener.cpp

- catkin_create_pkg july_listener std_msgs roscpp
- 在july_listener/src下编写代码
- 重点代码

  ros::Subscriber sub = n.subscribe(”july_listener“, 10, julyCallback); //声明一个监听器，和当收到这个topic发送的message之后要执行的操作

测试节点 & rqt_graph工具

- rosrun july_say july_say_node & rosrun july_listen july_listen_node
- Rostopic echo july_topic 查看控制台输出
- Rosrun rqt_graph rqt_graph 查看节点图



创建自己的message类型

- 创建一个叫july_msgs 的 package
- 添加 一个 int型参数 和 一个string参数
- 添加依赖到package.xml
  <build_depend>message_generation</build_depend>
  <exec_depend>message_runtime</exec_depend>
- 修改Cmakelist.txt

  find_package(catkin REQUIRED COMPONENTS roscpp rospy std_msgs message_generation)
  catkin_package(... CATKIN_DEPENDS message_runtime ... ...)
  add_message_files(FILES July.msg)
  generate_messages(DEPENDENCIES std_msgs)

- 重新编译
- 检查msg
  rosmsg show july_msgs/july

使用自己的message

- 对 CMakeList.txt 修改

  find_package(catkin REQUIRED COMPONENTS roscpp rospy std_msgs july_msgs)
  catkin_package(... CATKIN_DEPENDS july_msgs)

- 对 package.xml 修改

  <build_depend> july_msgs </build_depend>
  <exec_depend> july_msgs </exec_depend>

- 在源文件中引入 <july_msgs/JulyMsg.h>

使用ros参数

- n.param<参数类型>(参数名称, 参数变量, 参数默认值);
- 使用rosparam list 在控制台查看参数
- 使用rosparam get 在控制台获取参数
- 使用rosparam set 在控制台动态调整参数

Roslaunch

- 批量启动一系列节点
- 载入 rosparam 参数
- <node pkg=”包名" name=”节点昵称" type=”节点名"/>
- <param name=”参数名" value="参数值" />
- 使用launch

什么是 Docker

- Docker基于容器技术的轻量级虚拟化解决方案

- Docker是容器引擎，把Linux的cgroup、namespace等容器底层技术进行封装抽象， 为用户提供了创建和管理容器的便捷界面（包括命令行和API）

- Docker 是一个开源项目，诞生于 2013 年初，基于 Google 公司推出的 Go 语言实现

Docker 可以做什么

- 隔离系统环境
- 极简的安装和部署方式
- 让复杂系统安装配置成为历史

Docker 优点

1. 传统程序分发

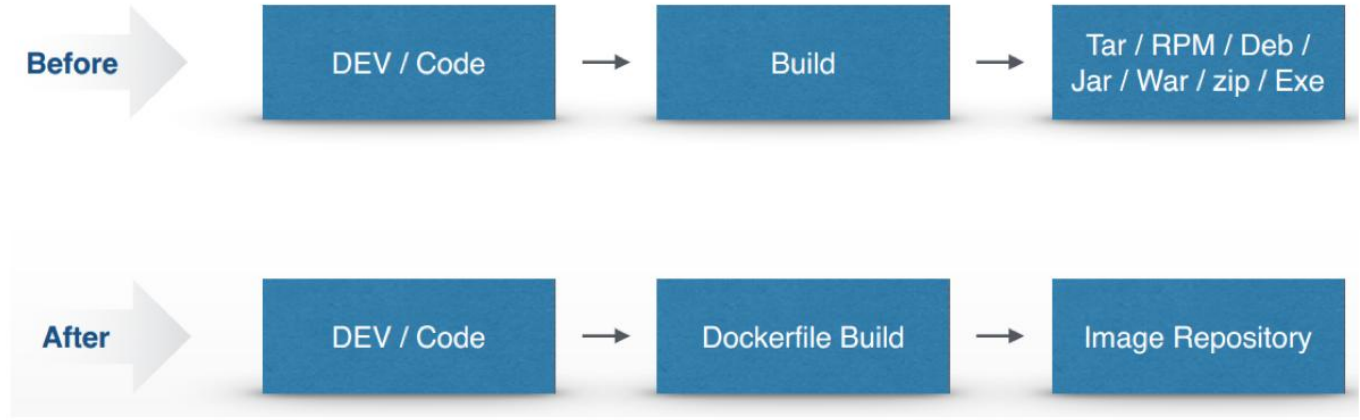
* 源代码分发： 需要用户自己编译，解决依赖问题

* 解决依赖时，对系统有侵入性，可能解决了这个依赖，其他的程序 又不能运行，python2 & python3等case

* 可执行文件分发： 多个平台多次编译，跨平台受限

2. docker分发

- 要求用户安装docker即可
- 环境，依赖均独立，不影响系统原有库



Docker 概念

- **Docker Images**

Docker image 是 Docker container 最基本的模板。image 通过容器使系统和应用易于安装，Docker image 是用来运行的容器，你可以在这里 https://hub.docker.com/找到许多 images（多种操作系统和软件已经被安装好了的 Docker）。

- **Docker Container**

Docker 容器（Docker Container）是一个 Image，在运行的 Docker image 上读取和写入。Docker 是一个联合的文件系统作为容器后台，容器的任何变化，都将被保存在一个基本 image 新的层上。我们安装应用程序的层就是容器。每个在主机上运行的容器都是独立的，因此，提供了一个安全的应用平台。

- **Docker Registry**

Docker registry 是为 Docker images 提供的库。它提供了公共和私有库。公共 Docker 库被叫做 Docker Hub。这里我们能够上传 push 和 pull 我们自己的 images。

Docker 安装使用

1. 安装Docker客户端

- https://yq.aliyun.com/articles/110806?spm=5176.8351553.0.0.53b11991Bpafu4

2. 从docker hub找到ROS docker

3. Docker pull 该镜像

使用docker运行july_say & july listener

1. 启动一个ros docker 客户端

   docker run -itd –v(源文件路径):(容器文件路径) --name (容器名) ros:kinetic

2. 进入july容器 docker exec –it 容器名 /bin/bash

3. 修改bashrc，

   echo ‘source /opt/ros/kinetic/setup.bash’ >> /root/.bashrc
   echo ‘source /catkin_ws/devel/kinetic/setup.bash’ >> /root/.bashrc

4. catkin_make & roslaunch

发布docker镜像

1. 使用阿里云hub
   http://cr.console.aliyun.com

2. 创建镜像仓库

3. Docker login 登录阿里云hub

4. Docker commit 容器名 阿里云仓库名:[镜像版本号]

5. Docker push 镜像

作业

1. 建立一个 talker listener模型
2. 构建自己的message，里面包含两个int32类型的值
3. Talker 通过该message发送随机两个数字，Listener收到该两个数字后累加后输出
4. 使用Docker打包上传，共享镜像

推荐材料

- Clion : https://www.jetbrains.com/clion/
- ROS官方tutorial: http://wiki.ros.org/ROS/Tutorials
- Gazebo& rviz
  http://wiki.ros.org/turtlebot_gazebo/Tutorials/indigo/Explore%20the%20Gazebo%20world

微信扫一扫关注我们

THANKS

张老师 无人驾驶高级算法工程师
https://www.julyedu.com/