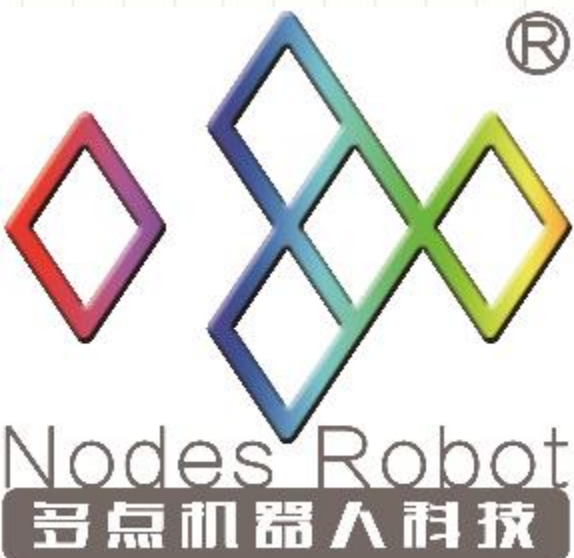




系统架构及概念



多点机器人

2017-06

目录

- ROS架构概念
- ROS框架
 - 开源社区级 The Community level
 - 文件系统级 The Filesystem level
 - 功能包
 - 综合功能包catkin / 堆rosbuild
 - 计算图级 The Computation Graph level
 - 节点
 - 消息
 - 主题
 - 服务



ROS架构概念

ROS框架1



ROS框架2



Linux/ROS

C++/
Python

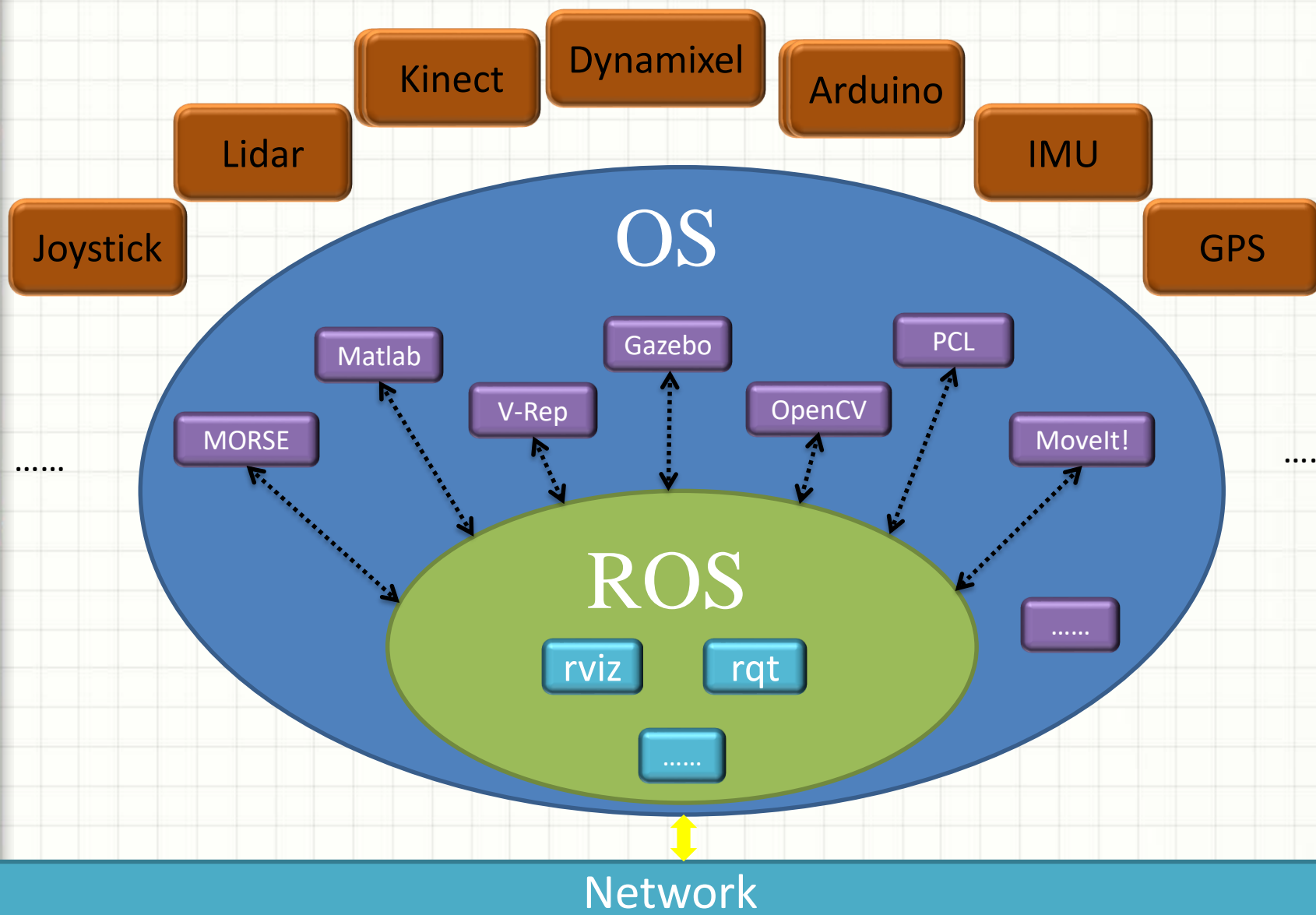
MCU/
ARM

Math/
Physics

AI/
VR



ROS框架3





ROS架构

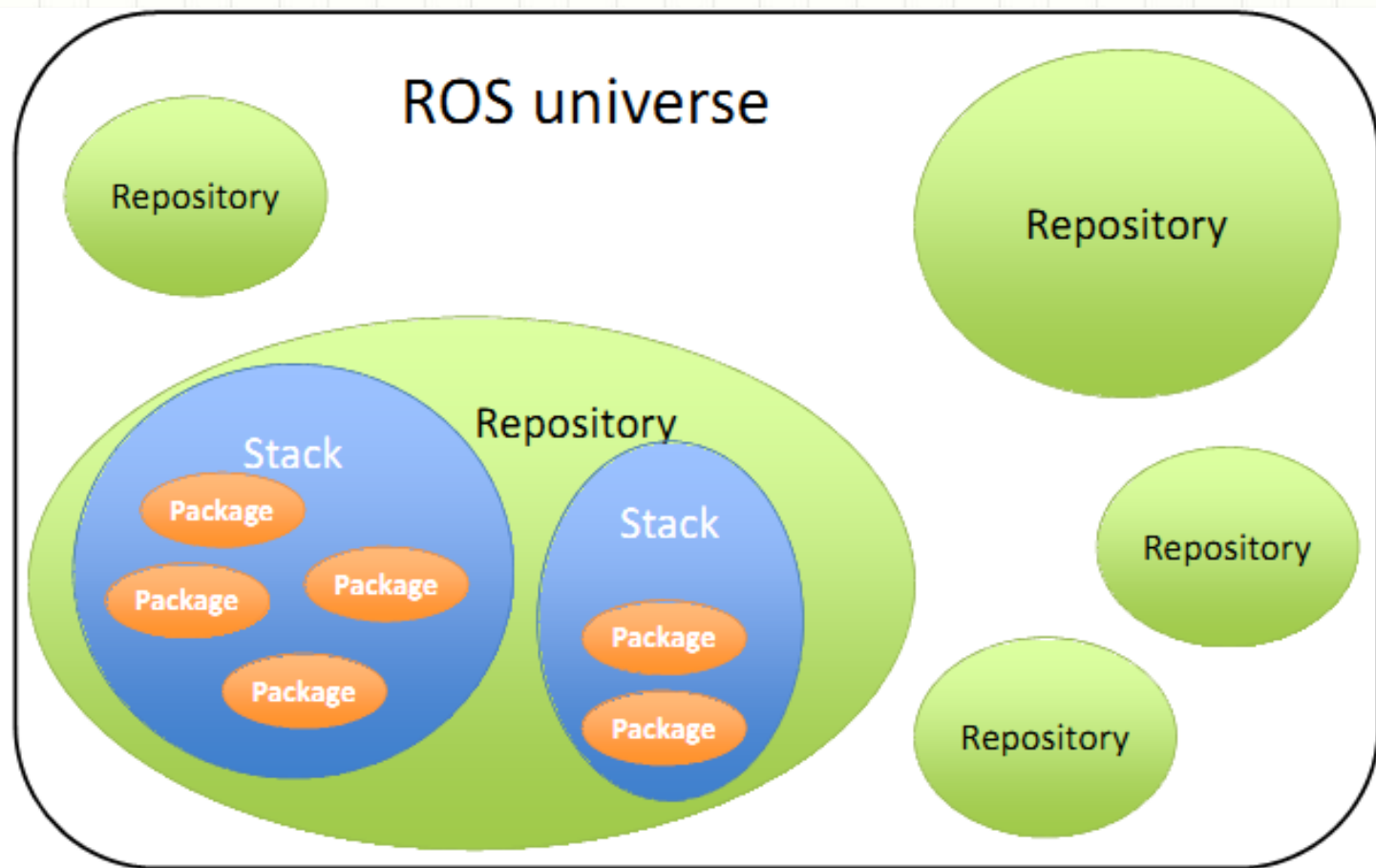
ROS系统架构

主要分为三个级别：

- 1 社区级 功能包的分布式管理
- 2 文件系统级 描述Node程序文件是如何组织和构建 catkin
- 3 计算图级 描述Node程序是如何通信和运行的

社区级

代码库的联合系统。使得协作亦能被分发。这种从文件系统级别到社区一级的设计让独立地发展和实施工作成为可能。正是因为这种分布式的结构，让ROS迅速发展，软件仓库中包的数量指数级增加。



文件系统级 catkin 架构

工作空间

{

build

devel

install

src/功能包 Package

{

功能包清单 [package.xml](#)

消息 msg

服务 srv

launch

其它

}

}

Package

Nodes

Messages

Services

Libraries

Tools

创建Catkin工作空间

- 查看当前系统的工作空间

```
$ echo $ROS_PACKAGE_PATH
```

- 建一个工作空间

```
$ mkdir -p ~/catkin_ws/src
```

```
$ cd ~/catkin_ws/src
```

```
$ catkin_make
```

- 应用配置

```
$ source devel/setup.bash
```

- 再次查看当前系统的工作空间 (注意: 只在本terminal有效)

```
$ echo $ROS_PACKAGE_PATH
```

文件系统级 功能包 ~/catkin_ws/src

- 创建一个ROS功能包mypkg

```
$ cd ~/catkin_ws/src
```

```
$ catkin_create_pkg pkgname rospy roscpp
```

指令

包名称 依赖项1 依赖项N

- 编译

```
$ cd ~/catkin_ws/
```

```
$ catkin_make
```

- 初始化工作空间

```
$ catkin_init_workspace
```

文件系统级 功能包 ~/catkin_ws/src

- CMakeLists.txt
- package.xml 清单, 关于功能包, 许可, 依赖, 编译等信息
- include/ : c++ header files
- src/ : 功能包源代码目录
- launch/ : launch file
- msg/ : message(.msg) types
- srv/ : service(.srv) types
- scripts/ : executable scripts

计算图级

计算图是ROS处理数据的一种点对点的网络形式。程序运行时，所有进程以及他们所进行的数据处理，将会通过一种点对点的网络形式表现出来。

这一级的重要概念：

节点 (node)、就是一个运算的进程模块，一个机器人系统通常有N个节点组成，例如：激光测距节点，电机控制节点，路径规划节点，图形界面节点，视觉处理节点，等等。

消息 (message)、节点之间通过C语言数据结构消息来通讯

主题 (topic)、消息以发布/订阅的方式传递，发布者和订阅者不了解彼此的存在，可以1对1，多对1，1对多，多对多的方式传递。

服务 (service)、一个**ROS控制器**，用于管理节点node。

ROS 1.0中，服务使用TCP的**ROS Master**来实现ROS控制器

ROS 2.0中，由**DDS**模型代替TCP的**ROS Master**

计算图级 实践

问题？

-