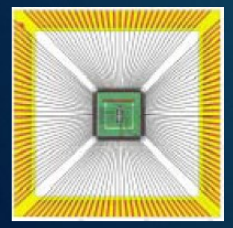
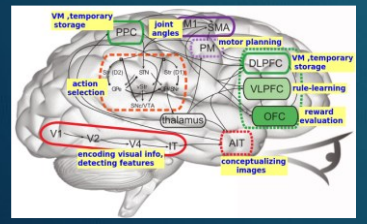
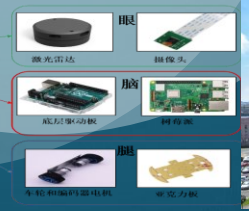
ROS数据记录、时间与模拟器

杰克科技 中央研究院




回顾与展望

- 什么是ROS
- ROS Workspace
- ROS 功能包与编译
- ROS节点
- ROS话题
- ROS消息与服务
- 消息发布与接收
- ROS Launch 文档
- rqt_graph等工具
- Turtlesim模拟
- ROS 参数服务器
- ROS namespace
- ROS 消息记录与回放
- ROS 时间
- Gazebo 模拟器

ROS Bags

- ROS bag 是存储消息的数据格式
- 其内容为二进制格式，以 .bag 作为文件扩展名
- 使用 ROS bag 文档存储消息数据，便于数据可视化及分析利用

将项目所有话题记录在bag文档中： `rosbag record --all` 或者 `rosbag record -a`

在bag文档中记录特定的话题： `rosbag record topic_1 topic_2 topic_3`

按 Ctrl + C 终止记录； 缺省情况下记录的文档名为开始记录的日期和时间；存储在当前目录

Rosbags:记录话题

```
mkdir ~/bagfiles
cd ~/bagfiles
rosbag record -a
```

```
rosbag record -O subset /turtle1/cmd_vel /turtle1/pose
```

-O 表示 rosbag record 把话题记录在名为 subset.bag 的文档中

rosbags：查看记录的消息

```
rosbag info <your bagfile>
```

year, date, and time and the suffix .bag

```
path:        2014-12-10-20-08-34.bag
version:     2.0
duration:    1:38s (98s)
start:       Dec 10 2014 20:08:35.83 (1418270915.83)
end:         Dec 10 2014 20:10:14.38 (1418271014.38)
size:        865.0 KB
messages:    12471
compression: none [1/1 chunks]
types:       geometry_msgs/Twist [9f195f881246fdfa2798d1d3eebca84a]
             rosgraph_msgs/Log   [acffd30cd6b6de30f120938c17c593fb]
             turtlesim/Color     [353891e354491c51aabe32df673fb446]
             turtlesim/Pose      [863b248d5016ca62ea2e895ae5265cf9]
topics:      /rosout                4 msgs    : rosgraph_msgs/Log   (2 connections)
             /turtle1/cmd_vel     169 msgs    : geometry_msgs/Twist
             /turtle1/color_sensor 6149 msgs  : turtlesim/Color
             /turtle1/pose        6149 msgs   : turtlesim/Pose
```

```
path:        subset.bag
version:     2.0
duration:    12.6s
start:       Dec 10 2014 20:20:49.45 (1418271649.45)
end:         Dec 10 2014 20:21:02.07 (1418271662.07)
size:        68.3 KB
messages:    813
compression: none [1/1 chunks]
types:       geometry_msgs/Twist [9f195f881246fdfa2798d1d3eebca84a]
             turtlesim/Pose      [863b248d5016ca62ea2e895ae5265cf9]
topics:      /turtle1/cmd_vel    23 msgs    : geometry_msgs/Twist
             /turtle1/pose      790 msgs    : turtlesim/Pose
```

Rosbags: 回放数据

```
rosbag play <your bagfile>
```

```
[ INFO] [1418271315.162885976]: Opening 2014-12-10-20-08-34.bag

Waiting 0.2 seconds after advertising topics... done.

Hit space to toggle paused, or 's' to step.
```

```
rosbag play -r 2 <your bagfile>
```

以2Hz的频率回放消息

rosbag 应用实例

Sensors Udacity Lincoln MKZ

Camera 3x Blackfly GigE Camera, 20 Hz

Lidar Velodyne HDL-32E, 9.5 Hz

IMU Xsens, 400 Hz

GPS 2x fixed, 1 Hz

CAN bus, 1,1 kHz

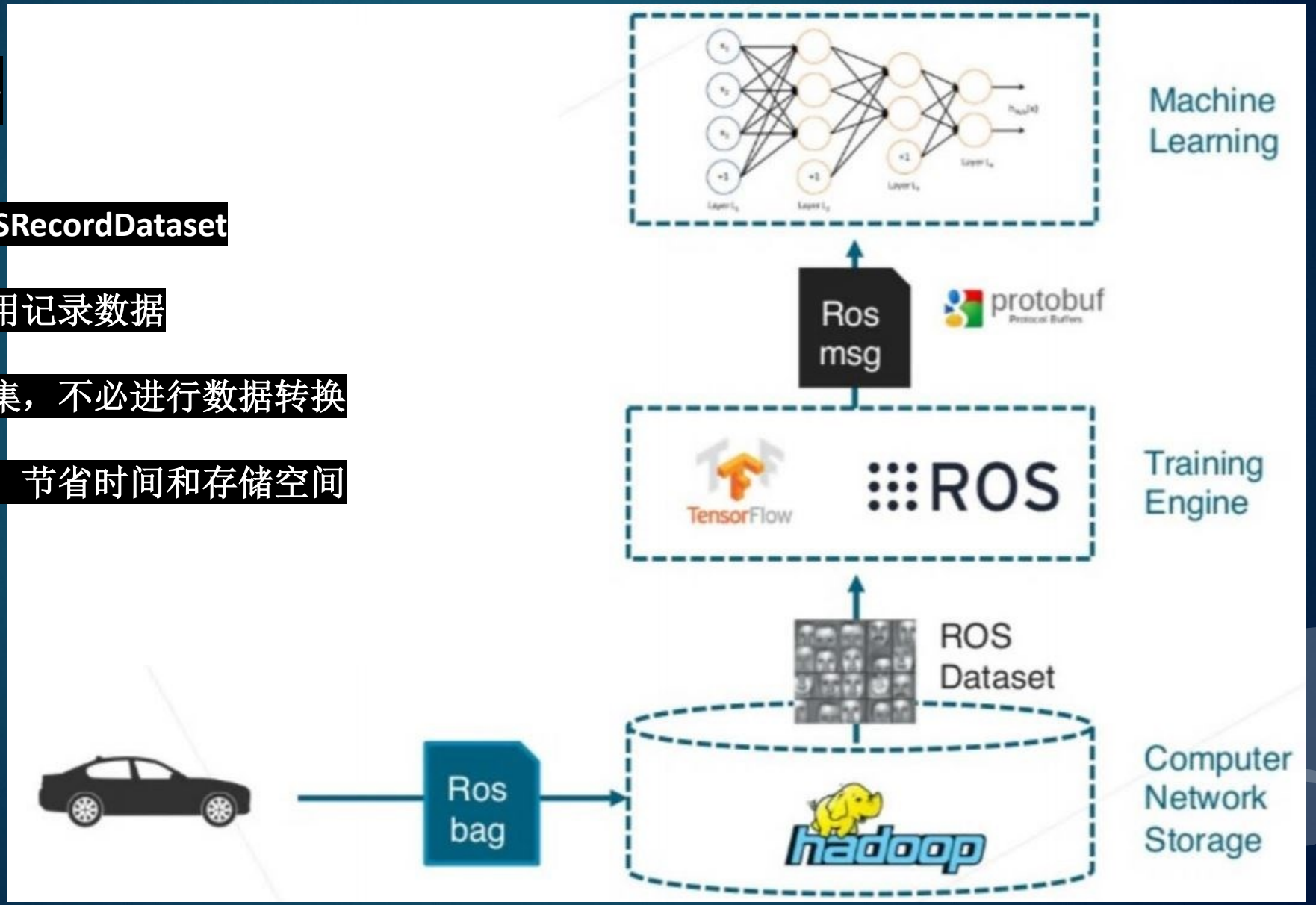
Robot Operating System

Data 3 GB per minute

rosbag 应用实例

训练神经网络

- Tensorflow ROSRecordDataset
- 存储并串行使用记录数据
- 直接使用数据集，不必进行数据转换
- 无需复制数据，节省时间和存储空间

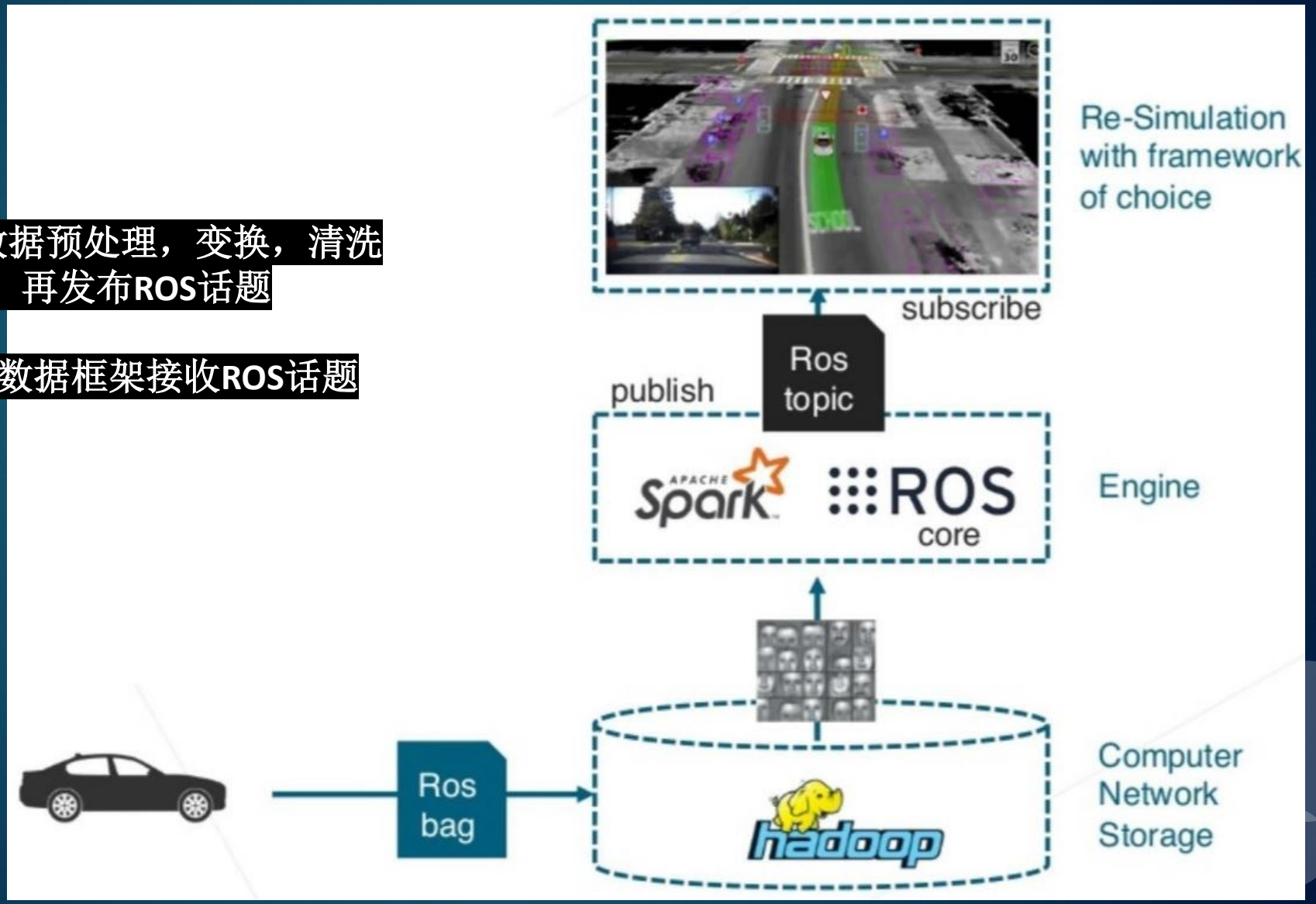


rosbag 应用实例

模拟与测试

- 可采用Spark进行数据预处理，变换，清洗聚类，等等操作后，再发布ROS话题
- 采用用户可选的大数据框架接收ROS话题

Autoware.org



Turtlesim 课堂练习：

mkdir ~/bagfiles

cd ~/bagfiles

roslaunch beginner_tutorials turtlemimic.launch

执行记录命令后，记得使用键盘操控乌龟

rosbag record –O filename -a

rosbag play filename.bag

ROS Time

Unix 时间

January 1st, 1970 00:00:00

- 以此为基准的相对时间，以秒为单位
- ROS消息的时间戳(timestamp)采用Unix时间

```
[INFO] [1581438700.596499]: hello world 1581438700.6
[INFO] [1581438700.597287]: /listener1I heard hello world 1581438700.6
[INFO] [1581438700.681260]: hello world 1581438700.68
[INFO] [1581438700.682367]: /listener1I heard hello world 1581438700.68
[INFO] [1581438700.696267]: hello world 1581438700.7
[INFO] [1581438700.696858]: /listener1I heard hello world 1581438700.7
[INFO] [1581438700.782215]: hello world 1581438700.78
[INFO] [1581438700.783369]: /listener1I heard hello world 1581438700.78
[INFO] [1581438700.796334]: hello world 1581438700.8
[INFO] [1581438700.797256]: /listener1I heard hello world 1581438700.8
[INFO] [1581438700.881575]: hello world 1581438700.88
[INFO] [1581438700.882349]: /listener1I heard hello world 1581438700.88
[INFO] [1581438700.896454]: hello world 1581438700.9
[INFO] [1581438700.897361]: /listener1I heard hello world 1581438700.9
[INFO] [1581438700.981365]: hello world 1581438700.98
[INFO] [1581438700.982205]: /listener1I heard hello world 1581438700.98
[INFO] [1581438700.997082]: hello world 1581438701.0
[INFO] [1581438700.997705]: /listener1I heard hello world 1581438701.0
```

ROS Time

- 通常， ROS采用计算机系统时钟作为其时间来源 （Wall Time）
- 在机器人模拟或回放数据时，也会用到模拟时间（simulated time)
- 如需使用模拟时间，进行如下参数设置

-- 设置 /use_sim_time 参数
% rosparam set use_sim_time true

-- 以下方法可将模拟时间发布到话题 /clock 上
* Gazebo (缺省为模拟时间）
* ROS bag (需使用选项 --clock)

时间和时长API： rospy.Time, rospy.Duration

- 可表示 wall time 或者 simulated time
- 数据结构相同

```
int32 secs
int32 nsecs
```

例如，获取当前时间：

rospy.Time.now(), rospy.get_rostime() Equivalent

```
1 now = rospy.get_rostime()
2 rospy.loginfo("Current time %i %i", now.secs, now.nsecs)
```

时间和时长API Sleeping: rospy.sleep(duration)

```
# sleep for 10 seconds
rospy.sleep(10.)

# sleep for duration
d = rospy.Duration(10, 0)
rospy.sleep(d)
```

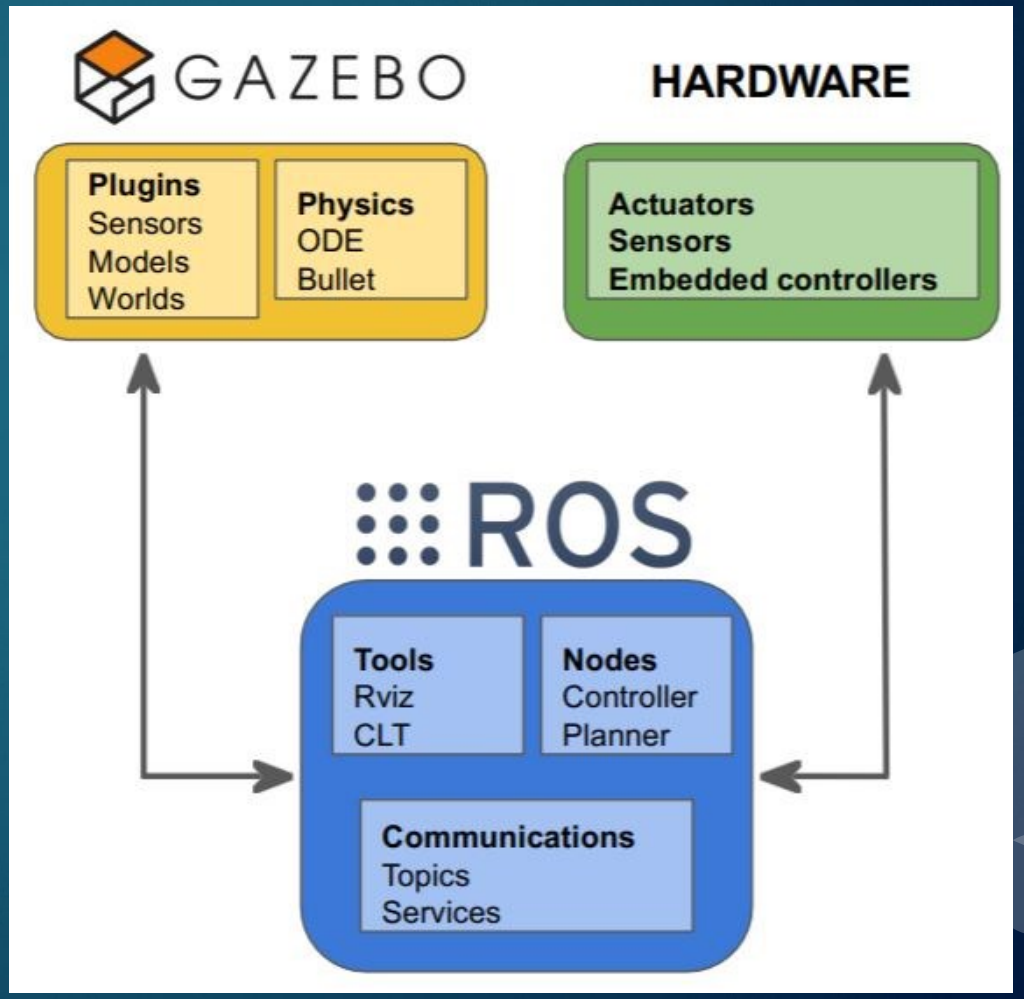
```
r = rospy.Rate(10) # 10hz
while not rospy.is_shutdown():
    pub.publish("hello")
    r.sleep()
```

详见 http://wiki.ros.org/rospy/Overview/Time

ROS模拟环境

- 模拟器被用来在某种程度上模仿真实的世界发生的事情
  - -- 在一个三维动态环境中模拟机器人、传感器或物体
  - -- 产生仿真的传感器反馈及物体之间的互动

- 为何需要仿真模拟
  - -- 省时省力
  - -- 非侵入或无损实验
  - -- 能模拟使用用户并没有的硬件
  - -- 生成专业的视频



模拟器架构

Gazebo模拟器同时运行以下二个过程 （processes）：

- 服务器：运行物理循环及生成传感器数据
  -- 可执行程序：gzserver
  -- 库：物理库、传感器库、数据提取回放、传输库

- 客户端：提供客户和模拟环境的互动及可视化功能
  -- 可执行程序：gzclient
  -- 库：传输、数据提取、GUI

分开运行Gazebo的服务器和客户端程序：

```
$ gzserver
$ gzclient
```

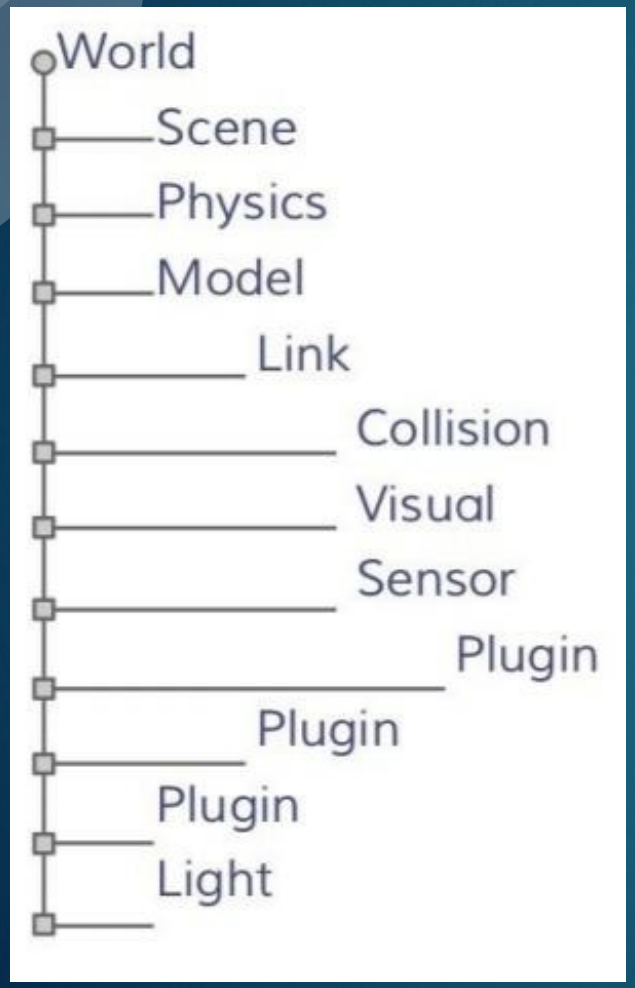
同时运行Gazebo的服务器和客户端程序：

```
$ gazebo
```

模拟要素

- 物理世界(world)：模型、光线、插件、物理属性等等集合
- 模型(model)：连接、关节、传感器 及插件
- 连接(link)：可视物体及其碰撞属性
- 碰撞物体(collision objects)：碰撞表的几何定义
- 可视物体(visual objects)：可视化表的集合定义
- 关节(joint)：连接之间的制约关系
- 传感器(sensor)：收集、处理及输数据
- 插件(plugin)：和环境、模型、传感或模拟器本身相关的程序

要素之间关系


World
Scene
Physics
Model
Link
Collision
Visual
Sensor
Plugin
Plugin
Plugin
Light


World
Insert
scene
physics
models
ground_plane
link
unit_box_1
link
lights
Property
Value
name
unit_box_1::link
self_collide
False
gravity
True
kinematic
False
pose
inertial
collision
unit_box_1::link::collision
visual
unit_box_1::link
visual
unit_box_1::link::visual

模拟物理环境(world)

- 模拟环境由层次化的模型组成
- Gazebo模拟器的服务器 (gzserver) 读入环境文档，以产生并建立环境

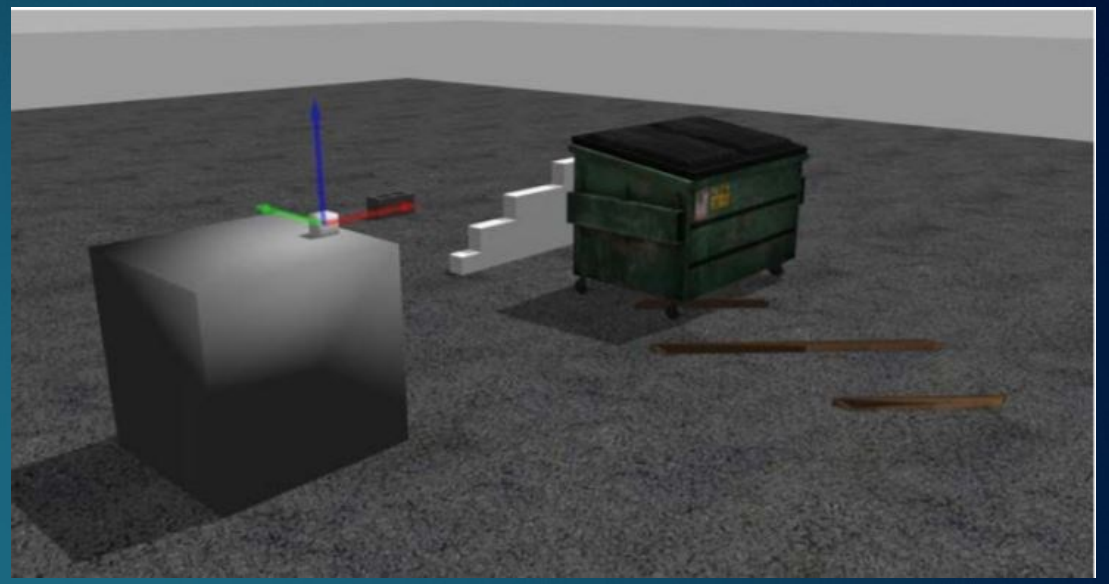
  -- 该环境文档格式为SDF(Simulation Description Format)或者URDF(Unified Robot Description Format)
  -- 具有 “*.world” 扩展名
  -- 包含一个模拟的所有要素，例如机器人，光线，传感器及静态物体等

模型 (model)

每一个模型均包含一些关键特点：

- 具体形象 (optional)：
  -- 模型体：圆球、立方体、组合外形等
  -- 运动学特点：关节、速度
  -- 动态：质量、摩擦、受力等
  -- 外观：颜色、纹理等

- 接口 (optional)：
  -- 控制与反馈接口 (libgazebo)



要素种类

- 碰撞与可视外观
  (Collision and Visual Geometry)

  -- 简单形状：球形、圆柱体、立方体
  平面等
  -- 复杂形状：高度图、网状图等

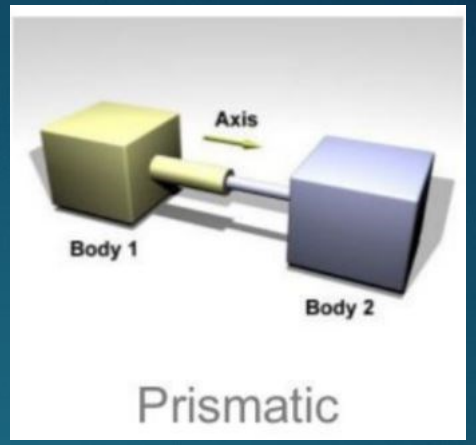
要素种类

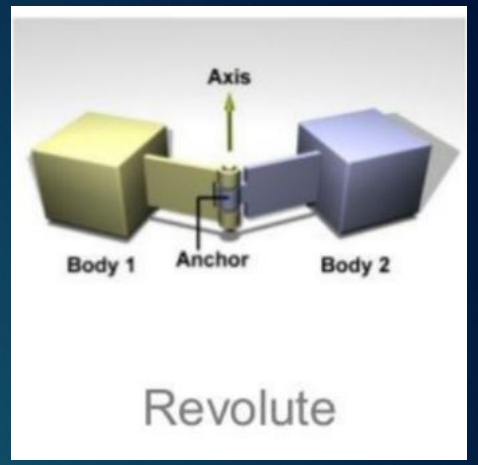
- 碰撞和可视物体

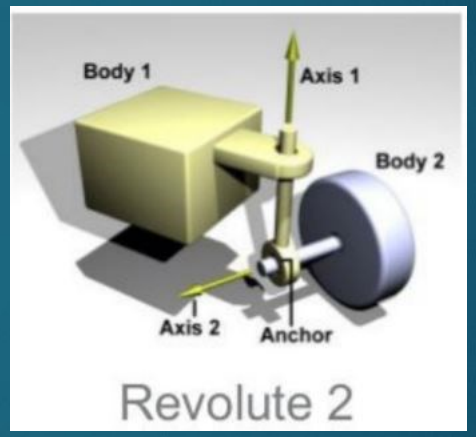
  -- 简单形状：圆球、圆柱、立方体、平面
  -- 复杂形状：高度图、网状图

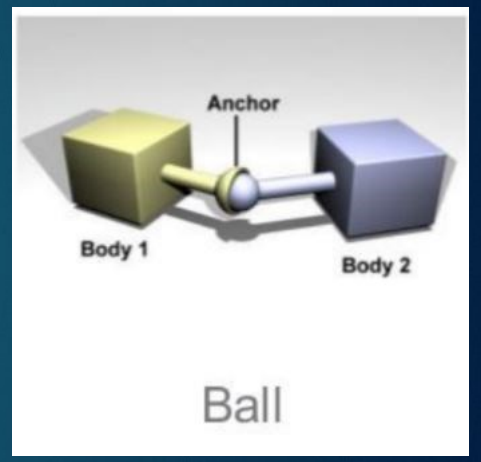
- 关节

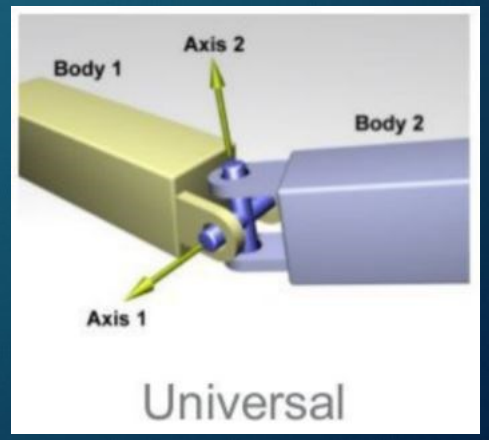
  -- 滑动（Prismatic）： 一个自由度平移
  -- 旋转 （Revolute): 一个自由度旋转
  -- 旋转2 （Revolute2): 二个旋转关节相连
  -- Ball: 三个自由度旋转
  -- Universal: 二个自由度旋转
  -- Screw: 一个自由度平移加上一个自由度旋转











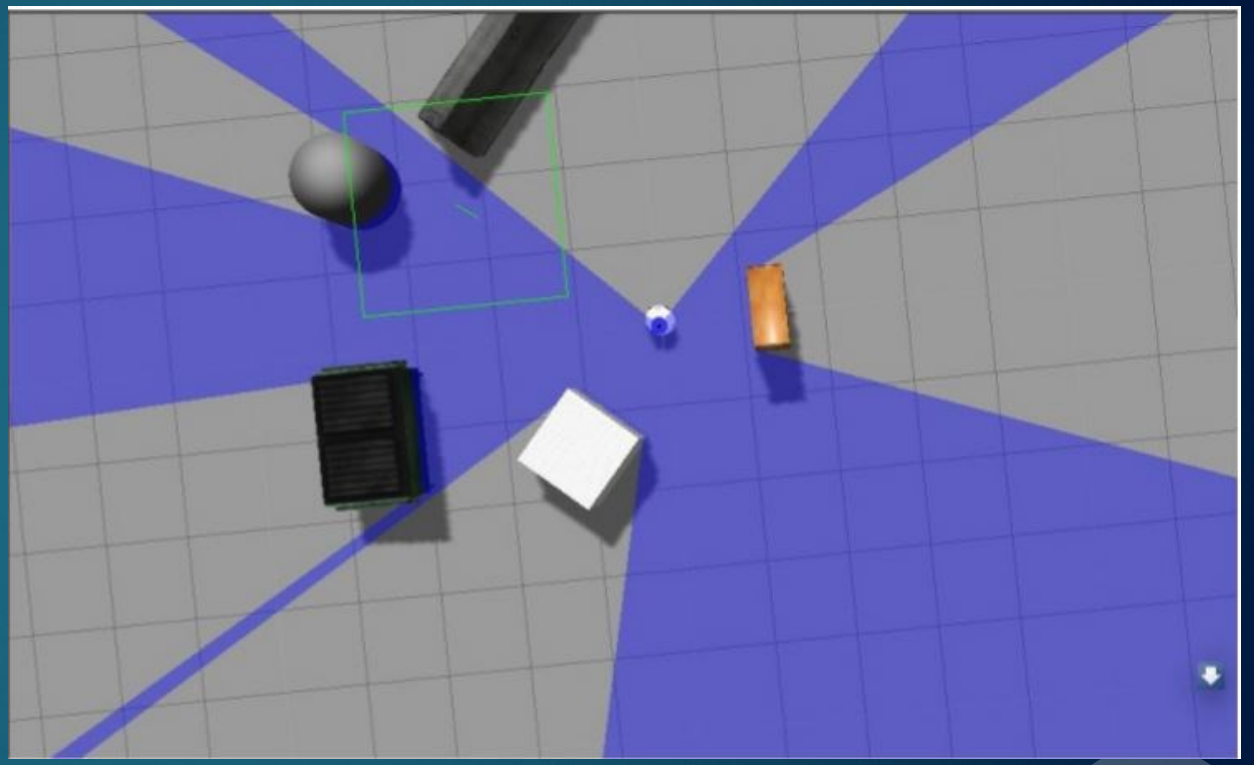
要素种类

- 传感器：

  -- 光线（ray)：产生周边环境数据
  -- 相机 （2D与3D）：产生图像或景深数据
  -- 接触：产生碰撞数据
  -- RFID：检测RFID标签

- 光亮(light)：

  -- 点光源（Point）：全向光源，例如灯泡
  -- 发散光源（Spot）：定向发散光源，
  如舞台灯光
  -- 定向光源：平行定向光源，例如日光



如何采用Gazebo模拟机器人

Steps:

1. load a world
2. load the description of the robot
3. spawn the robot in the world
4. publish joints states
5. publish robot states
6. run rviz