

# 基于 ROS 的云机器人服务框架<sup>①</sup>

陈 贤<sup>1,2</sup>, 武延军<sup>3</sup>

<sup>1</sup>(中国科学院软件研究所 互联网软件技术实验室, 北京 100190)

<sup>2</sup>(中国科学院大学, 北京 100190)

<sup>3</sup>(中国科学院软件研究所 基础软件国家工程研究中心, 北京 100190)

**摘 要:** 随着机器人技术的发展和硬件的普及, 机器人的市场应用前景越来越广泛. 但是, 目前的机器人仍存在着许多局限, 具体表现为可移植性弱, 本地计算成本高, 服务应用较少等. 云机器人的提出, 将机器人的计算能力从本地迁移到云端, 不但提高了机器人的计算能力, 降低了机器人硬件成本, 而且能使资源的分配更为均衡, 为解决机器人发展面临的困难提供了有效的解决途径. 在云机器人的基础思想之上, 提出一种基于 ROS 的云机器人服务框架. 该服务框架使用开源的机器人操作系统 ROS 作为机器人运行的基础, 增强了其对于不同硬件和软件环境的可移植性. 同时, 在框架的云端部分加入了机器人的服务管理系统和服务解析模块, 能够方便快速地对机器人服务进行扩展和调用. 在最后的实验部分, 通过人脸识别服务模块对云机器人服务框架进行了实验验证.

**关键词:** 机器人; 云机器人; 机器人操作系统; 云机器人服务框架

## Service Framework for ROS-Based Cloud Robot

CHEN Xian<sup>1,2</sup>, WU Yan-Jun<sup>3</sup>

<sup>1</sup>(Laboratory for Internet Software Technologies, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China)

<sup>2</sup>(University of Chinese Academy of Sciences, Beijing 100190, China)

<sup>3</sup>(National Engineering Research Center of Fundamental Software, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China)

**Abstract:** With the development of robot technology and the popularity of hardware, the prospect of robot market is much brighter than in the past. However, there are still many limitations of robots, in particular the weakness of portability, high computing costs and few applications. With the raise of cloud robotics, computing process can be migrated from local robot to cloud, not only improves the computing power and reduces hardware costs of robot, but also allows a more balanced allocation of resources. In this paper, it proposes a service framework for ROS-based cloud robot. This service framework uses open-source robot operating system ROS as the basis for running the robot, enhancing its portability to different hardware and software environments. Meanwhile, it joins the service management system and service analysis module in the cloud, so that robot service can be easily expanded and quickly called. As the experiment part of the paper, it uses a face recognition service module to testing our framework for ROS-based cloud robot.

**Key words:** robot; cloud robotics; ROS; robot service framework

自从 1959 年“机器人之父”恩格尔先生发明第一台机器人以来, 人类相继设计出各式各样的机器人. 相对于人类, 机器人在多个方面有着明显的优势, 目前

许多工厂已经使用机器人来代替人类劳动力进行作业, 一些服务岗位也开始引入机器人为人类提供服务. 伴随着现代科学技术的飞速发展, 硬件和软件成本大幅

① 基金项目: 中国科学院先导专项(XDA06010600); 国家自然科学基金重大项目(91218302)

收稿时间: 2016-01-13; 收到修改稿时间: 2016-02-26 [doi:10.15888/j.cnki.csa.005343]

度降低,机器人的普及程度正处于直线上升的趋势。众多的数据和迹象表明,机器人的市场需求量正在不断上升,应用的范围越来越广,将是未来几年研究和发展的热点领域。

机器人虽然有着广阔的市场前景和飞速的技术发展,但从相关技术来看,无论在硬件还是软件方面都尚未成熟。本文试图解决机器人在软件方面遇到的一些困难和挑战。首先,绝大多数的机器人必须使用特定型号的操作系统和应用软件来作为其运行的基础,这样虽然能够保证系统的健壮性,但无疑在极大程度上阻碍了机器人的普及。其次,单机的机器人为了保证轻便和灵活性,同时避免昂贵的造价,往往不会搭载过于强大的计算能力。而随着机器人应用场景的不断复杂,尤其是对图像声音等处理功能的加入,强大的计算能力已经必不可少。另外,与所有智能硬件平台刚形成初期类似,机器人平台也面临着应用单一,服务开发困难等问题。没有足够服务来支撑的平台,其发展速度通常非常缓慢。

本文针对目前的机器人在软件方面面临的几个问题,使用开源的机器人操作系统,设计了一套用于机器人数据计算和服务管理的云机器人服务框架,用于解决机器人可移植性弱、本地计算成本高、服务应用稀缺等方面的问题。

## 1 相关工作与研究现状

### 1.1 云计算和云机器人

2006年8月9日,Google首席执行官埃里克·施密特(Eric Schmidt)在搜索引擎大会(SES San Jose 2006)首次提出“云计算”(Cloud Computing)<sup>[1]</sup>的概念。在传统的云计算平台即服务PaaS、软件即服务SaaS、基础设施即服务IaaS的基础上,很快就有学者提出了机器人即服务RaaS<sup>[2,3]</sup>。在RaaS的基础上,来自卡耐基梅隆大学的James Kuffner教授第一次引入了“云机器人”的概念<sup>[4]</sup>,引起了广泛的关注和讨论<sup>[5,6]</sup>。“云机器人”的引入,打破了原先机器人的发展格局,将机器人和云计算这两块热门领域进行更加紧密的结合,使得机器人在社会普及和功能实现上有了更多的可能性,势必成为机器人和云计算领域中的下一个热点。

结合当前云计算的应用场景,云机器人的优势主要可以体现在以下的几个方面:

更强大的计算能力。如前文所言,单机的机器人

为了保证轻巧性,同时避免过于昂贵的造价,往往不会搭载强大的计算能力。而随着机器人应用场景的不断复杂,尤其是对图像声音等处理功能的加入,强大的计算能力已经必不可少。基于云计算的机器人可以将本地复杂的数据计算、规划决策操作迁移到云端,通过原始数据的上传和对结果的采集,即可快速地地完成复杂的功能,例如有学者在2010年提出的DAvinCi机器人云计算框架<sup>[7]</sup>。

更合理的资源分配。机器人作为一种硬件资源,与其他资源一样,其使用率存在着峰值和谷值。此模型强调机器人作为行使服务的个体,使用云计算对机器人进行分配和调度,将机器人资源和服务存储在云端,机器人硬件设备可以快速地注册到云端,建立一个机器人的集合<sup>[8]</sup>,云端的管理系统也可以很好根据使用者的需求对机器人资源进行合理的分配。在这种模型下,强调机器人是完成服务的个体,整个系统可以分为物理资源层、资源管理层和服务层。机器人在资源层注册后,使用者就可以通过服务层来获取想要的服务,资源管理层会根据请求的服务和机器人资源的状态来进行合理的调度。

更智能的服务。云计算使得各个机器人个体之间不是在相互隔离的,而是可以互相地学习和合作。例如RoboEarth计划<sup>[9,10]</sup>,这个计划由来自全世界著名学校和公司的35名科学家共同启动,旨在建立机器人自己的资料库和互联网。这种模型下,系统主要由一个云端数据库和多个具备学习能力的个体组成,机器人通过在云端的信息共享和相互学习,独立或者联网协作完成工作<sup>[11]</sup>。

### 1.2 ROS和ROS\_bridge

ROS(Robot Operating System)<sup>[12]</sup>是由Willow Garage公司在2010年发布的一个开源的机器人操作系统,目前最新版本为ROS 2.0 alpha版本。ROS旨在简化机器人开发在软件方面的代价、提高代码的复用性和模块化标准,在发布之初就受到了机器人领域学者的的大力支持。ROS提供了不同硬件的抽象描述、底层驱动程序的管理、程序节点间消息传递的机制、机器人软件发行包的管理等功能,使用统一的节点管理框架,从而保证机器人能够按照程序的设定,顺利协调地完成工作。

ROS\_bridge<sup>[13]</sup>是ROS开源环境下的一个扩展模块,由Christopher Crick, Graylin Jay, Sarah Osentosiki,

Benjamin Pitzer 和 Odest Chadwicke Jenkins 提出。ROS\_bridge 是一个运行在 ROS 上的中间件,它能够支持非 ROS 环境下的设备与 ROS 环境的设备进行相互通信,使得 ROS 在控制和开发方面更加容易和高效。通过 ROS\_bridge,可以在一个没有安装 ROS 环境的设备,例如手机等移动设备上,对安装了 ROS 系统的机器人下达命令。

## 2 服务框架设计

### 2.1 云机器人服务框架特点

针对于云计算和机器人相结合的环境,云机器人的服务架构必须具备以下几个特点:

**高效性:** 云机器人设计的初衷就是为了提高服务运行的速度,因此云机器人服务框架在引入额外模块的同时必须保证运行效率不受影响。通过简化数据传输的步骤,优化服务模块的运行流程,可以保证服务运行的高效性。

**可移植性:** 由于目前机器人的硬件和软件技术发展尚未成熟,没有一套被业界广泛认可的标准,导致各个公司生产的机器人存在着硬件标准和运行环境的不统一,因此针对机器人开发的架构需要一定的可移植性。可移植性要求云机器人的海端可以被快速地移植到不同的软硬件环境下,云端可以被方便地部署到不同服务器环境中。本文将 ROS 作为海端机器人运行的环境,解决机器人的可移植性问题。

**可扩展性:** 机器人服务应用的数量正在不断增长,机器人需要不断地更新服务来执行更多的功能,因此云机器人必须拥有较高的可扩展性。可扩展性要求新的可执行服务能够快速部署在云端,以便于海端的机器人进行调用。本文在云端设计了一个服务注册管理系统,使得服务的提供者只要按照一定的规则便能快速地部署和更新服务。

**实时性:** 不同于其他的系统,机器人产生的数据具有较高的实时性。在 ROS 中,传感器产生的消息会不断地发送到对应的话题上,因此从话题上获取的消息都是传感器最新采集到的消息,系统数据的实时性也得到了相应的保证。

### 2.2 云机器人服务框架整体设计

云机器人服务架构主要分为云端(服务端),海端(机器人本地端)以及海云通信协议三个部分。云端部分主要包括云计算所需的硬件资源(例如存储空间、内

存、处理器等)、云计算架构支持(例如 HBase、Map-Reduce、HDFS 等<sup>[14]</sup>)以及服务层的服务主体和管理配置模块,完整的云端是云机器人架构提供服务的保障。海端部分主要涉及机器人硬件和操作系统,是机器人能够顺利运行的保障。云端和海端之间通过海云通信协议进行消息的传输。

在目前云机器人模型中,基于 SOA<sup>[15]</sup>服务框架的模型是一种常见的模型,其模型架构主要如图 1 所示。此模型将云端划分为硬件资源层、云计算支持层和 SOA 服务层<sup>[16,17]</sup>,并在海端使用定制化的操作系统和 SOA 框架本地端,海端和云端之间使用传统的通信协议进行消息的传输。此模型由传统的 SOA 服务模型演变而来,适用于拥有大规模云服务和标准化机器人客体的情况。这类模型的缺点也非常明显:需要定制化的机器人操作系统,需要 SOA 服务框架对云端和海端的支持,而且没有突出海端机器人的特性。

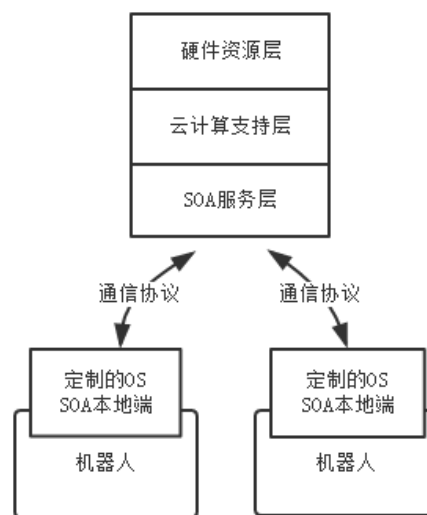


图 1 基于 SOA 服务框架的云机器人架构图

本文在 SOA 模型的基础上,使用 ROS 来代替定制化的操作系统,并在云端使用特定的服务层来代替原 SOA 的服务层,用于解决机器人的兼容问题,降低开发和维护成本。由于 ROS 本身拥有一套适合机器人行为描述的通信协议,因此可以简化对机器人控制的步骤。在这些基础上,本文提出了双端 ROS 的云机器人服务架构,如图 2 所示。

为了进一步简化框架的复杂性,本文通过在海端环境中加入 ROS\_bridge 模块,从而消除云端对 ROS 环境的依赖,最终形成了如图 3 的架构。这种架构功

能层次更加简明, 搭建更加方便, 运行也更加高效.

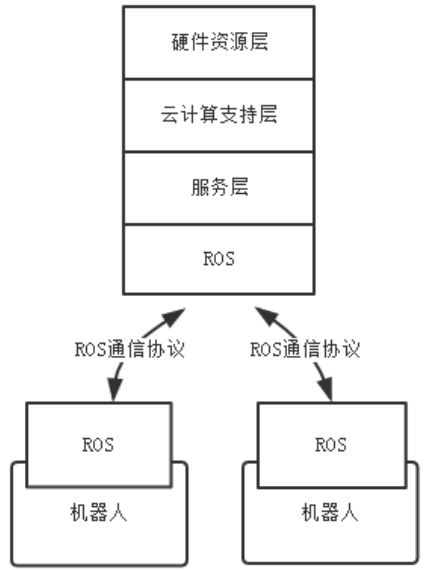


图2 基于双端 ROS 的云机器人服务架构图

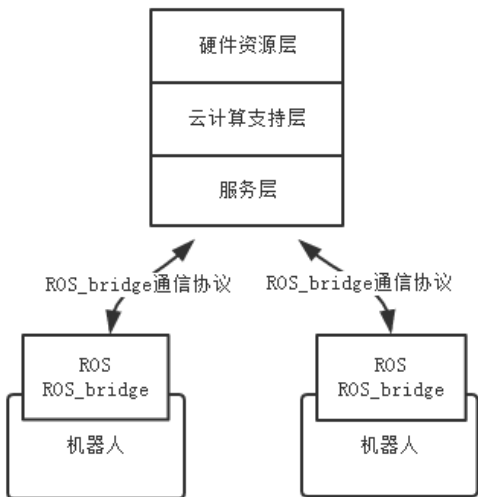


图3 使用 ROS\_bridge 的云机器人服务架构图

### 2.3 云端服务层框架设计

云端服务层为机器人提供必要的服务, 是云机器人能够顺利完成任务的保障, 也是整个系统框架的核心部分. 从服务对象的角度, 服务层提供的服务主要针对海端机器人和服务提供者两个方面.

对于海端机器人, 服务层提供了请求解析、服务调用和结果反馈等功能. 如图 4 所示, 其核心模块主要有服务收发端、服务解析模块、服务反馈模块、服务管理系统、服务模块和云计算支持层. 服务收发端用于从海端接收传输过来的消息并传递给服务解析模块, 同时也可以将服务运行的结果反馈给海端机器人.

服务解析模块主要对海端传输过来的服务请求消息进行解析, 根据解析后的请求完成对服务的调用. 服务反馈模块的功能是将服务模块运行的结果回传给海端的机器人. 服务管理系统主要负责服务的配置和分发. 服务模块和云计算支持层指服务的提供者在云计算环境下为海端机器人提供的可执行服务, 例如语言控制、人脸识别等.

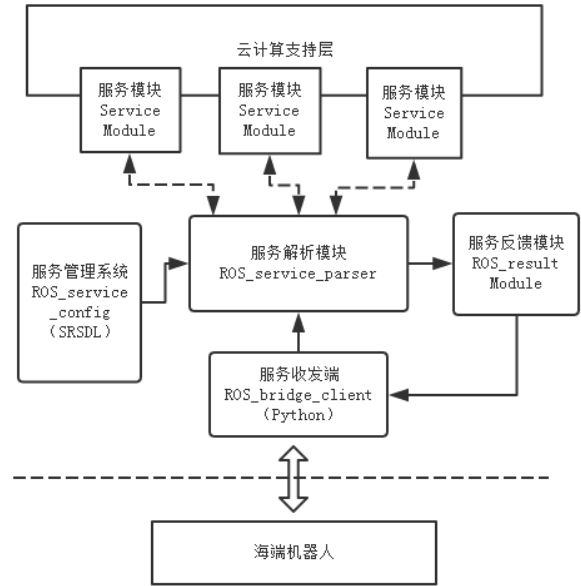


图4 针对海端机器人的云端服务层架构图

对于服务的提供者, 服务层提供服务注册、修改、删除等常用功能. 如图 5 所示, 其核心架构分为服务管理模块、服务配置模块、服务云端存储、服务日志系统和服务分发五个方面. 服务管理模块主要接收服务提供者的注册、修改和删除请求. 服务配置模块主要用于管理服务的配置信息, 包括文件路径、调用参数和运行方式等. 服务云端存储提供了海量的存储空间用来存储服务的主体文件和运行所需的知识库文件. 服务日志系统用来记录服务管理模块产生的日志, 用于维护和管理. 服务分发是指服务层在得到外部请求时对服务进行分发和运行, 以完成具体的功能.

## 3 模块详细设计

### 3.1 使用 Python 实现云端消息的接收和发送

ROS\_bridge 提供了两种与 ROS 环境设备进行通信的方式, 一种是基于 web 的 socket 技术, 称为 ROSJS<sup>[18]</sup>, 另一种是采用 JSON 格式封装的消息传输

机制. ROSJS 适合于网页端对机器人的控制, 而云端的服务层需要使用封装的 JSON 库来与 ROS 设备进行通信. 为了方便海端使用 ROS\_bridge 和云端进行快速的消息交互, 本文在 JSON 格式消息传输的基础上, 使用 Python 封装了一套能够快速接收和发送 ROS 消息的接口, 称其为 ROSPY, 如图 6 所示.

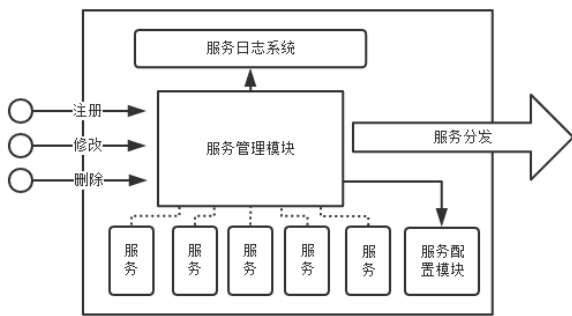


图 5 针对服务提供者的云端服务管理系统

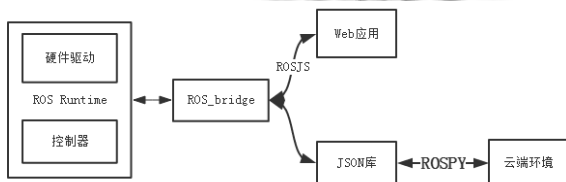


图 6 ROSPY 模块示意图

ROS\_bridge 定义了十分复杂的消息格式, 而 ROSPY 使得开发人员无需考虑这些复杂的消息格式, 使用面向对象的思想, 快速地实现非 ROS 环境与 ROS 环境之间的通信交互功能. 目前, ROSPY 支持与 ROS\_bridge 的连接、大数据的传输、话题建立、话题发送、话题接收、服务建立、服务调用、服务反馈等功能, 并根据不同的海端机器人内置了一些常用的配置, 实现与 ROS 环境高效的交互.

### 3.2 服务管理系统和机器人服务描述语言

针对于 ROS 系统的特点, 本文在服务层自定义了一种描述机器人服务的语言格式. 每一个注册到云端的服务都需要使用该格式的语言定义后才能被正确调用. 服务描述语言使用键值对的形式描述一个具体的服务, 其中包括了服务名称, 开发者, 模块路径, 服务版本号, 参数列表, 结果列表, 知识库路径, 日志路径和服务描述等. 服务层中人脸识别服务的描述语言如图 7 所示.

云端的服务由服务管理系统进行统一的管理和控制. 海端应用不需要知道云端服务运行时所需的参数,

因为这些参数由传感器产生, 并且已经自动发布到了对应的话题节点上. 调用服务所需的参数信息都从相应的话题上获取, 结果也将以话题的形式进行发布. 相比于传统模式, 机器人服务描述语言具有易维护和易扩展等优点.

```

{
  service_name = 'faceapp',
  developer = 'chenxian',
  module = 'hello_face.py',
  version = 1.1
  paras = [
    '/face_image%std_msgs/String',
    '/time%std_msgs/String',
  ],
  results = [
    '/face_result%std_msgs/String',
  ],
  knowledge = 'face.dat'
  log = '/log/face.log',
  desc = 'This is an app for face detect and identify.'
}

```

图 7 人脸识别服务的描述语言

### 3.3 服务解析模块

云端接收了来自海端的数据后, 需要将数据进行解析, 并根据解析的结果对服务进行分发, 这部分功能由服务解析模块完成. 服务解析的步骤主要包括: 提取服务名、读取服务配置、获取参数、调用服务和发布结果等几个步骤, 具体流程如图 8 所示.

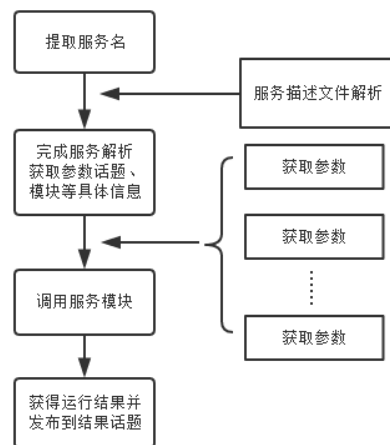


图 8 服务解析流程图

提取服务名: 海端机器人将服务名使用字符串的形式发布在特定的话题(ros\_service)上, 服务解析模块通过 ROSPY 获取话题(ros\_service)上的数据, 并从字符串中解析提取服务名.

读取服务配置: 云端根据提取到的服务名, 通

过服务管理模块,从中获取到服务的相关信息和配置.主要包括运行参数的话题信息、反馈结果的话题信息和服务运行的主体.

获取参数:云端通过 ROSPY 来获取相应话题上的参数.因为传感器数据不停地被发布到话题上,所以获取到的参数总是最新的.参数获取使用异步通信的方式进行,主模块无需额外的等待.

调用服务:调用服务运行的主体是服务解析模块的最终目的.云机器人的本质,就是将服务从本地迁移到云端,利用云端强大的资源,减少服务运行的时间.

发布结果:服务主体将结果以标准输出的方式返回,并通过 ROSPY 发布到结果话题上.海端的机器人收到话题上的数据后,即可做出相应的行为.

### 3.4 可扩展的服务模块

服务模块是进行运算任务的模块,主要是一些需要复杂计算或者有着较大知识库的应用,例如人脸的识别、声音的控制、图像的处理等.

为了对云机器人进行验证,本文编写了一个运行在云端的人脸识别服务模块.人脸识别模块可以通过从对应的话题上读取图像信息,然后在云端进行识别,并返回识别结果.与本地的人脸识别比较,基于云机器人服务框架的人脸识别服务增加了图像传输的过程,并将海端的主要任务移到云端进行.其流程主要如图9所示.

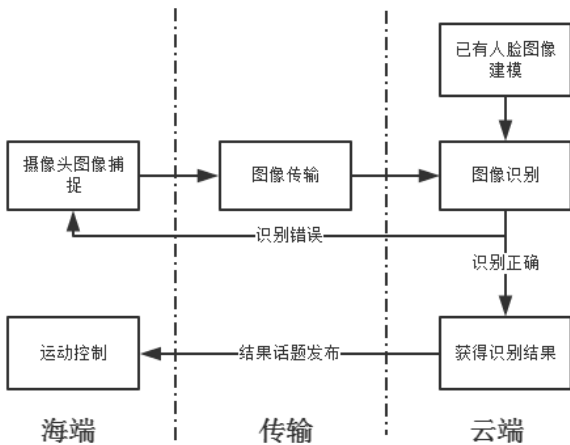


图9 人脸识别流程图

## 4 实验部署与验证

为了验证云机器人服务框架的可行性,测试云机

器人平台的效果,本文在常见的硬件平台上进行了实验部署.本实验中采用的机器人为开源的 TurtleBot2 移动机器人平台<sup>[19]</sup>,其搭载了普通的双核笔记本作为运行核心,并且已经安装 ROS 和 ROS\_bridge 环境.本实验的云端环境采用了拥有 i7 四核 CPU 和 16G 内存的台式机进行模拟,在其之上部署云机器人服务框架.实验使用服务层的人脸识别服务模块进行测试和结果收集.

通过实验我们可以得到了模块运行期间各部分的时间消耗,由于图像抓取和识别成功率等因素,人脸识别需要连续识别多次图像后才能确认人脸,按照实验中识别成功所需平均次数为 4 次进行计算,最终的时间如表 1 所示.

表 1 云机器人模型下人脸识别运行时间(单位:秒)

	单次耗时	模块耗时
人脸建模	9.393	9.393
图像传输	0.094	0.376
图像识别	0.551	2.204
结果发布	0.00013	0.00013
总时间	/	11.973

进一步分析实验,发现影响该实验结果的因素主要有 人脸建模效率、图像识别效率、图像传输效率、图像识别准确率几个方面.结合云机器人服务框架的特点,可以将人脸识别服务模块在几个方面进行优化:

云端的人脸建模可以提前进行,或者和第一次图像传输同时进行.

在图像识别结果出来前预先进行下一次图像传输,由于图像传输所需时间小于图像识别所需时间,从而能够保证在多个图像识别的过程中不需要进行额外的等待.

优化后的运行流程如图 10 所示.

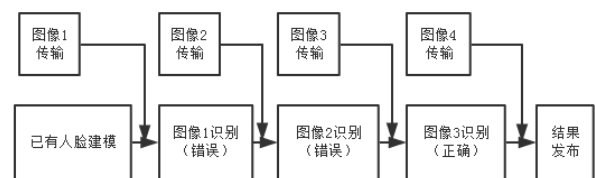


图 10 优化后的人脸识别模块流程

最后,通过实验比较人脸识别服务模块在单机模块、云机器人模型和云机器人优化模型的运行效率,得到如表 2 的结果.



表 2 三种模型下运行时间比较(单位: 秒)

实验环境	单机模式	云机器人	云机器人优化后
人脸建模	12.788	9.393	9.108
图像传输	/	0.094 (0.376)	/
图像识别	0.707 (2.828)	0.551 (2.204)	0.590 (2.360)
发布	/	0.0001	0.0001
总耗时	15.616	11.973	11.468

上述实验结果是在识别次数固定为 4 次的前提下得出的。当识别次数发生变化时, 三种模型所需时间也将发生变化, 具体如图 11 所示。

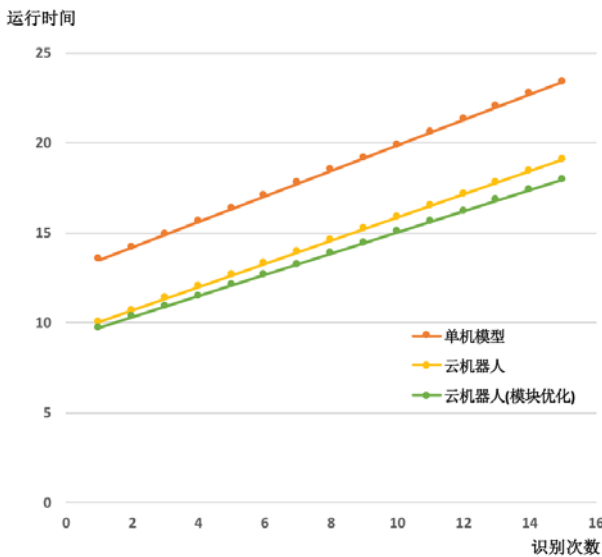


图 11 人脸识别时间随次数的变化(单位: 秒)

从实验结果看出, 图像传输所需的时间远小于人脸建模和图像识别所需时间, 且随着识别次数的增加, 人脸识别服务在云机器人模型下的花费时间依然小于单机模型。其主要原因是  $T_0 < \Delta T$ , 其中  $T_0$  为图像传输所需时间,  $\Delta T$  为云模型下运行服务相比单机模型下节省的时间。另外, 因为实验涉及到网络的传输, 因此不同网络环境会对实验结果产生一定的影响, 当然如果没有网络, 也就不存在云机器人模型一说。

## 5 结语

基于 ROS 的云机器人服务框架是针对机器人单机环境下所面临的问题, 结合云计算环境提出的一种机器人服务模型。本文在传统的机器人云计算框架基

础上进行改进和创新, 使用开源机器人操作系统 ROS 作为机器人的运行环境, 并在云端加入服务层响应海端 ROS 设备的服务请求。

在最后, 本文进行了实验设计和部署, 结果表明此服务框架在确保可移植性和可扩展性的同时, 相比单机运行模型有着更高的执行效率, 从而验证了基于 ROS 的云机器人服务框架的可行性和有效性。

## 参考文献

- 1 Armbrust M, et al. Above the clouds: A Berkeley view of cloud computing. Eecs Department University of California Berkeley, 2009, 53(4): 50-58.
- 2 Schaffer HE. X as a service, cloud computing, and the need for good judgment. IT Professional, 2009, 11(5): 4-5.
- 3 Chen YN, Du ZH, Marcos GA. Robot as a service in cloud computing. 2010 Fifth IEEE International Symposium on Service Oriented System Engineering (SOSE). IEEE. 2010.
- 4 Kuffner J. Robots with their heads in the clouds. Discovery News, 2011.
- 5 Goldberg K, Kehoe B. Cloud robotics and automation: A survey of related work[Tech. Report]. EECS Department, University of California, Berkeley. 2013, UCB/EECS-2013-5.
- 6 Hu GQ, Wee PT, Wen YG. Cloud robotics: Architecture, challenges and applications. Network, IEEE, 2012, 26(3): 21-28.
- 7 Arumugam R, et al. DAViNCi: A cloud computing framework for service robots. 2010 IEEE International Conference on Robotics and Automation (ICRA). IEEE.
- 8 Du ZH, et al. Design of a robot cloud center. 2011 10th International Symposium on Autonomous Decentralized Systems (ISADS). IEEE. 2011.
- 9 Zweigle O, et al. RoboEarth: Connecting robots worldwide. Proc. of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human. ACM. 2009.
- 10 Waibel M, et al. A world wide web for robots. IEEE Robotics & Automation Magazine, 2011, 18(2): 69-82.
- 11 Kamei K, et al. Cloud networked robotics. Network, IEEE 2012, 26(3): 28-34.
- 12 Quigley M, et al. ROS: An open-source robot operating system. ICRA Workshop on Open Source Software. 2009, 3. (3).

- 13 Crick C, et al. Rosbridge: Ros for non-ros users. Proc. of the 15th International Symposium on Robotics Research. 2011.
- 14 Taylor RC. An overview of the Hadoop/ MapReduce/ HBase framework and its current applications in bioinformatics. BMC Bioinformatics 11, 2010(Suppl 12): S1.
- 15 Erl T. Service-oriented architecture: concepts, technology, and design. Pearson Education India, 2005.
- 16 Remy SL, Blake MB. Distributed service-oriented robotics. Internet Computing, IEEE, 2011, 15(2): 70–74.
- 17 Chen YN, Bai XY. On robotics applications in service-oriented architecture. 28th International Conference on Distributed Computing Systems Workshops, 2008. ICDCS'08. IEEE. 2008.
- 18 Osentoski S, et al. Robots as web services: Reproducible experimentation and application development using rosjs. 2011 IEEE International Conference on Robotics and Automation (ICRA). IEEE. 2011.
- 19 Garage W. Turtlebot. <http://turtlebot.com>. [2011-11-25].

[www.c-s-a.org.cn](http://www.c-s-a.org.cn)

[www.c-s-a.org.cn](http://www.c-s-a.org.cn)