

**本科毕业设计（论文）开题报告**



|  |  |
| --- | --- |
| **题目：** | **基于kubernetes的** |
| **serverless函数服务系统的设计与实现** | |

|  |  |  |  |
| --- | --- | --- | --- |
| **学生姓名** | **母先涛** | **学　号** | **201731064212** |
| **教学院系** | **计算机科学学院** | | |
| **专业年级** | **物联网工程2017级** | | |
| **指导教师** | **杨明根** | **职　称** | **工程师** |
| **单　　位** | **计算机科学学院** | | |

**目录**

[1 绪论 1](#_Toc516078158)

[1.1 选题的背景和意义 1](#_Toc516078159)

[1.2 设计的目的和内容 2](#_Toc516078160)

[1.3 选题的技术现状 2](#_Toc516078161)

[2 选题的主要内容 3](#_Toc516078162)

[2.1 任务概述 4](#_Toc516078163)

[2.2 设计的总体结构描述 4](#_Toc516078164)

[2.3 设计的预期结果 4](#_Toc516078165)

[3 设计的技术路线 5](#_Toc516078166)

[3.1 开发工具和开发环境 5](#_Toc516078167)

[3.2 重点与难点分析 5](#_Toc516078168)

[4 设计的时间进度安排 6](#_Toc516078169)

[5 已有的参考文献 7](#_Toc516078170)

# 绪论

## 选题的背景和意义

云计算涌现出很多改变传统IT架构和运维方式的新技术，比如虚拟机、容器、微服务，无论这些技术应用在哪些场景，降低成本、提升效率是云服务永恒的主题。过去十年来，我们已经把应用和环境中很多通用的部分变成了服务。Serverless的出现，带来了跨越式变革。Serverless把主机管理、操作系统管理、资源分配、扩容，甚至是应用逻辑的全部组件都外包出去，把它们看作某种形式的商品——厂商提供服务，我们掏钱购买。过去是“构建一个框架运行在一台服务器上，对多个事件进行响应”，Serverless则变为“构建或使用一个微服务或微功能来响应一个事件”，做到当访问时，调入相关资源开始运行，运行完成后，卸载所有开销，真正做到按需按次计费。这是云计算向纵深发展的一种自然而然的过程。

Serverless是一种构建和管理基于微服务架构的完整流程，允许你在服务部署级别而不是服务器部署级别来管理你的应用部署。它与传统架构的不同之处在于，完全由第三方管理，由事件触发，存在于无状态（Stateless）、暂存（可能只存在于一次调用的过程中）计算容器内。构建无服务器应用程序意味着开发者可以专注在产品代码上，而无须管理和操作云端或本地的服务器或运行时。Serverless真正做到了部署应用无需涉及基础设施的建设，自动构建、部署和启动服务。国内外的各大云厂商 Amazon、微软、Google、IBM、阿里云、腾讯云、华为云相继推出Serverless产品，Serverless也从概念、愿景逐步走向落地，在各企业、公司应用开来。

在现阶段，Serverless主要应用在以下几个场景。首先在Web及移动端服务中，可以整合API网关和Serverles服务构建Web及移动后端，帮助开发者构建可弹性扩展、高可用的移动或 Web后端应用服务。在IoT场景下可高效的处理实时流数据，由设备产生海量的实时信息流数据，通过Serverles服务分类处理并写入后端处理。另外在实时媒体资讯内容处理场景里，用户上传的音视频到对象存储OBS，通过上传事件触发多个函数，分别完成高清转码、音频转码等功能，满足用户对实时性和并发能力的高要求。无服务器计算还适合于任何事件驱动的各种不同的用例，这包括物联网，移动应用，基于网络的应用程序和聊天机器人等。Serverless架构让人们不再操心运行所需的资源，只需关注自己的业务逻辑，并且为实际消耗的资源付费。可以说，随着Serverless架构的兴起，真正的云计算时代才算到来了。

## 设计的目的和内容

目前serverless技术分两类：FaaS和BaaS。此设计要完成的是一个实现基于kubernetes的serverless的函数服务系统，即FaaS系统的实现。主要还是要搭建在基于底层一点的kubernetes上面，并实现kubernetes的集群部署FaaS的函数功能。多个集群节点都可以调用同一个函数服务。

## 选题的技术现状

### 容器技术的兴起

对于很多用户而言，云和非云环境并不是二选一，而是两者都需要。云和非云环境在相当长的一段时间内将同时存在。应用如何快速地在云和非云环境中迁移成为一个重要的问题。为了有更高的可用性，避免厂商的锁定，一些实力雄厚的客户往往同时是多家云平台供应商的客户。因此实现应用在不同云环境中的快速迁移也成为一个重要的需求

容器（Container）技术的出现为前文提及的问题提供了一个很好的解决方案。经过几年的迅速发展，容器已经不容置疑地成为云计算的一项关键基础技术。Docker（现在已经更名为Moby项目）已成为容器引擎的事实标准。Kubernetes也在竞争中脱颖而出，成为容器编排（Orchestration）平台的事实标准。通过Kubernetes这样的容器编排平台，容器镜像可以快速地被部署到成百上千的主机上。Kubernetes成为一种类似操作系统的存在，有的人认为Kubernetes就是一种云操作系统。传统的操作系统只管理一台主机上的CPU、内存、磁盘和网络资源，而Kubernetes则掌控着数据中心中成百上千台主机的资源。

### **Serverless与容器**

容器架构中最小的运行单元是容器，而Serverless中则是函数。容器应用一般是预先部署，然后持续在线。而在Serverless架构中，应用是按需加载和执行的。这意味着理论上Serverless的资源使用效率更高。

其实，容器技术可以是Serverless架构实现的一个基础。容器平台的最小运行单元为容器，虽然目前容器内一般运行的是一个完整的应用，但是将容器内运行的对象变成函数显然并无技术困难。Kubernetes上默认没有事件触发的支持，无法做到按需部署容器应用。但是通过Kubernetes叠加上一些FaaS框架运行包含函数逻辑的容器，用户很容易使Kubernetes具备FaaS服务的能力，容器必将是未来私有云构建Serverless能力的一个重要实现基础。

**Serverless实现的基础**

函数即服务（Function as a Service，FaaS）是当前Serverless实现的技术基础。每一个函数完成一个相对简单的业务逻辑，一个完整的应用由若干个函数组成。因为FaaS和Serverless的关系密切，因此FaaS的特点同时也是Serverless平台的特点：

1.抽象了底层计算

2.资源按使用量付费

3.自动弹性扩展

4.事件驱动

FaaS是当前Serverless实现的重要基础，所以有一部分人认为Serverless就是FaaS。这是狭义上的Serverless。

目前的开源serverless框架：

* [fission](https://github.com/fission/fission) - Fast Serverless Functions for Kubernetes [http://fission.io](http://fission.io/)
* [knative](https://github.com/knative) - Kubernetes-based platform to build, deploy, and manage modern serverless workloads.
* [kubeless](https://github.com/kubeless/kubeless) - Kubernetes Native Serverless Framework [http://kubeless.io](http://kubeless.io/)
* [openfaas](https://github.com/openfaas/faas) - OpenFaaS - Serverless Functions Made Simple for Docker & Kubernetes
* [openwhisk](https://openwhisk.incubator.apache.org/) - Apache OpenWhisk (Incubating) is a [serverless](https://en.wikipedia.org/wiki/Serverless_computing), open source cloud platform that executes functions in response to events at any scale.

等等。。。。。。

# 选题的主要内容

1. Serverless的了解、学习
2. docker、kubernetes的学习与安装部署
3. 开源框架OpenFaaS学习，并部署在kubernetes上面。
4. 系统搭建好了以后写一个简单函数服务的运行调用。

## 任务概述

使用容器技术，部署kuernetes镜像集群，再在多台服务器部署的集群上面搭建FaaS函数服务系统镜像，编写函数服务，实现多节点的访问和调度。

## 设计的总体结构描述

首先需要多台服务器，然后选择一个服务器做主节点（master），其余的服务器做从节点（node1、node2.....）。具体部署kunernetes集群在这几台服务器上面：1.在所有的服务器上面安装kubernetes三大组件（kubeadm，kubelt，kubectl）与docker。2.在选择一个服务器作为master，部署kubernetes Master。在master服务器上面初始化kubernetes集群。3.每个节点都安装Pod网络插件（flannel插件）。4.其他从节点加入master节点初始化的集群中来。5.在master节点部署OpenFaaS框架。6.编写一个函数服务。

## 设计的预期结果

毕业设计预期得到成果如下：

● 毕业设计开题报告

● 毕业设计任务书

● 毕业设计日志

● 实现基于kubernetes的serverless函数服务系统设计

● 毕业设计论文

# 设计的技术路线

OpenFaaS 、docker、kubernetes、Sunny-Ngrok

## 开发工具和开发环境

操作系统：Vmware/CentOS7

程序语言：Java、Go

开 发 包：JDK1.8、OpenFaaS开源框架

开发平台：IDEA

## 重点与难点分析

serverless的难点：

1.不适合长时间运行应用。Serverless 在请求到来时才运行函数，这意味着，当应用不运行的时候就会进入 “休眠状态”，当请求再次来临时，应用需要一个启动时间如果应用需要长期不间断的运行、处理大量的请求，那么就不适合采用 Serverless 架构

2.完全依赖于第三方服务。当用户采用 Serverless 架构的时候，就和特定的服务供应商绑定了，假如用户使用了AWS的Serverless服务，由于兼容性问题，当他再想把应用迁移到阿里云上时，可能会增加很多额外的成本

项目难点：

kubernetes集群的部署，master节点必须是2核cpu以上，还有就是多台服务器。使用的是虚拟机技术，在虚拟机上面划分多个虚拟服务器。使用yaml文件部署，需要填写不同版本号，还有就是必须安装网络插件flannel，不然节点状态会一直是NotReady，启动不了。对本机的网络状况还是很有要求的，很多资源都是在github上面拉取的，如果网络连通不畅很可能拉取失败。不同版本的kubernetes有不同的yaml配置文件，切记要下载对应的版本。因为在本地安装搭建的，要穿透到公网上面去，需要内网穿透技术Sunny-Ngrok。最后就是函数的编写，构建推送和部署到OpenFaas上面。

## 3.3.设计的时间进度安排

|  |  |  |
| --- | --- | --- |
| **序号** | **设计（论文）各阶段内容** | **起止日期** |
| 1 | OpenFaaS的学习 | 2021年3月5日-3月15日 |
| 2 | docker、kubernetes的学习 | 2021年3月16日-4月17日 |
| 3 | kubernetes搭建OpenFaaS | 2018年4月18日-5月20日 |
| 4 | 论文撰写 | 2021年5月20日-6月4日 |
| 5 | 准备答辩 | 2021年6月6日-6月15日 |

# 

# 4.已有的参考文献

1. OpenFaaS实战：https://xinchen.blog.csdn.net/article/details/109805296
2. kubernetes集群部署 https://www.cnblogs.com/heian99/p/12173599.html
3. OpenFaaS简化无服务器功能 <https://github.com/openfaas/faas>
4. Serverle介绍https://github.com/serverless/serverless/blob/master/README\_CN.md
5. OpenFaaS源码阅读 <https://zhuanlan.zhihu.com/p/41056766>
6. kubernetes中文指南 https://www.lanqiao.cn/library/kubernetes-handbook/concepts/pod-overview