



上海交通大学

SHANGHAI JIAO TONG UNIVERSITY

高压数字测量系统课程设计

## 开 题 报 告

姓名                      谢弘洋

学号                      515021910641

同组姓名              孟诗涵    陈晓彤

2018 年 10 月 7 日

## 一 设计目标

本次高压数字测量系统课程设计围绕赵老师提出的广域分布互感器智能网联技术展开，本组主要围绕其中的云服务器部分进行，并完成以下设计目标：

1. 搭建云服务器平台，安装及配置服务器框架、资源和端口。
2. 实现云服务器通过网络端口收发数据与命令的功能，与 4G 模块进行通讯。
3. 实现云服务器处理、分析数据数据的功能。
4. 以 web 应用的形式实现网页、手机客户端，进行数据访问和监控。
5. 实现云服务器以 61850 通信规约向电力系统专网传输数据的端口。

综合以上设计目标，能够实现云服务器与互感器数据采集终端、客户端和电力系统专网三者的通讯，向互感器数据采集终端发送命令以读取波形和状态数据，应用云服务器的计算能力完成波形数据的处理和分析，监测广域范围内的电力设施运行状态；另一方面，通过 web 客户端和电力系统专网能够实现对服务器中历史数据的访问，以及通过云服务器向广域数据采集终端发送指令，实现对广域范围内互感器的远程监测。云服务器既作为分布广泛的各个数据采集终端的大脑，又作为沟通不同用户和电力系统数据之间的桥梁，在广域分布互感器智能网联技术发挥重要作用。

图1展示了云服务器的基本结构以及在整个广域分布互感器智能网联技术项目中发挥的作用。

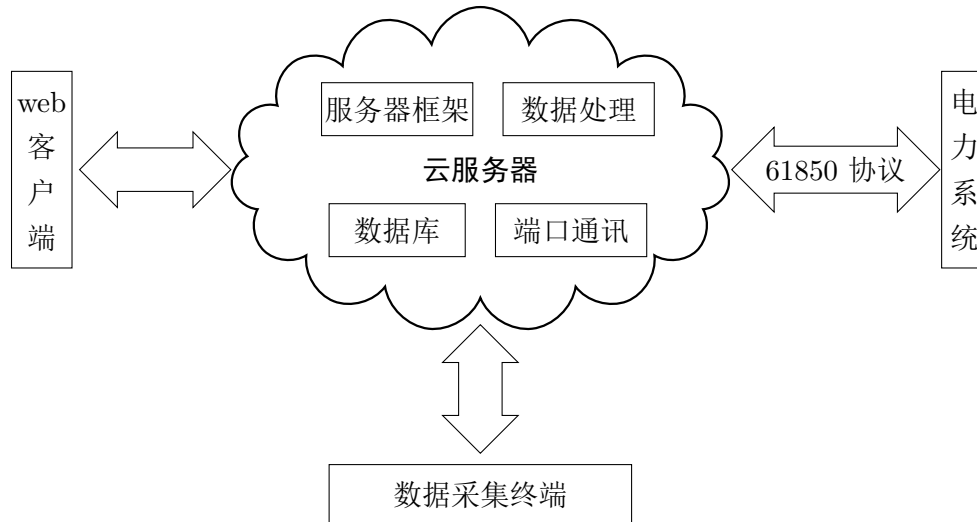


图 1: 设计目标结构示意图

## 二 初期思路

在服务器项目的编写上，选择以较为熟悉且应用广泛的 Python 为主，结合设计目标，服务器端的项目基本结构如图2所示。其中，端口通讯部分完成和数

据采集终端之间的指令发送和数据接收功能，数据处理部分可以对接收的波形进行进一步分析处理，web 应用作为服务端程序，实现和不同客户端之间的交互，而数据库负责存储历史数据，将各个功能模块联系在一起。

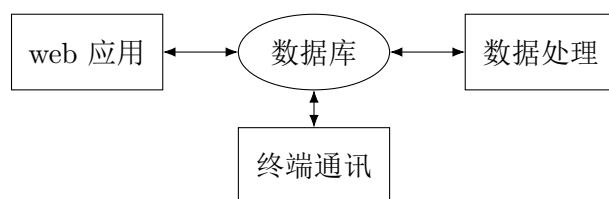


图 2: 服务器端结构示意图

### 1. 终端通讯

终端通讯部分可以通过 TCP 协议和各个数据采集终端通过 TCP Socket 套接字进行通讯，使用 socket 库实现 Socket 实例的创建，并为其指定地址和端口号，建立和终端之间的 TCP 连接，之后即可进行双向通讯，该方法实现简单，在服务器端只需要一个 TCP 通讯线程实现数据收发，在数据采集终端只需要使用 4G 模块 AT 指令集的 TCP 连接命令即可实现通讯，并且服务器端也可以随时像数据终端发送指令，但由于采用长连接方式，每个数据终端需要占用服务器的一个端口。

另外也可以采用 Http 请求的形式实现数据的传输，该方式相对直接通过 TCP 连接的方式更为灵活，可以复用服务器端口资源，也可以直接在 web 应用中实现数据的接受，但是增加了数据采集终端的编程难度，同时，由于 http 每次请求结束后连接即断开，因此需要采用附加的机制如 Ajax 或 Comet 来实现服务器端的主动通讯，往往占用大量资源或提高程序的复杂度。

### 2. 数据处理

数据处理部分可以直接使用 Python 代码实现，前期可以做一下简单的运算以验证服务器的数据处理能力。而针对项目的实际应用功能，需要进行数字滤波，并通过 FFT 运算计算得到频率、有效值、相位等信息，以及根据实际需要据此进行运行状态的分析与判断，这些功能均可以利用 Python 丰富的资源来实现。

### 3. web 应用

#### web 框架

web 应用程序围绕着接受 Http Request，根据 Request 内容进行相应处理，并返回 Http Response 的形式展开，使用 web 框架的目的在于免除应用开发者进行重复性的枯燥劳动，而可以专注于完成处理 Request 的部分。

在 Python web 下主流的框架有 Flask 和 Django 两种，其中，Flask 更为灵活、自由，适合于小型项目开发，整体性能优于 Django；而 Django 更为成熟、完善，但整体较为冗杂，灵活度较低，适合企业与团队项目。结合两者特点，计划首先通过 Flask 框架轻量、灵活的特点熟悉 web 应用的基本架构，并探索一些基本功能的实现；在项目后期结合实际情况，如果 Flask 框架足以胜任，则继续使用，如果需要转换到 Django 框架，也可以利用其上手快，第三方库丰富的特点快速地完成项目迁移。

#### web 接口

对于不同的 web 服务器和 web 框架之间，需要建立同一的接口使得各种框架和服务端之间能够顺利通讯，在 Python 下，WSGI（Web Server Gateway Interface，Web 服务器网关接口）实现了这一功能，因此可以使用 Gunicorn 或 Nginx/uWSGI 等来运行 Python web 应用。

### 4. 数据库

对于数据类型较少，结构简单的项目，可以直接使用轻量的关系数据库 SQLite，直接在服务器端本地建立数据库，并且 Python 以及 Flask 框架均对其有良好的支持，可以快速建立并应用。如果需要更为完善的数据库服务，腾讯云也提供了 MySQL 服务器，并采用按量计费的方式使用。

### 5. 手机客户端开发

随着 HTML5 的发展，从 web 应用向手机 APP 的迁移和整合越来越简便，可以通过开发简单的手机 APP 框架后嵌套 web 应用的方式，将客户端从网页迅速迁移到手机上，免除单独开发手机应用程序的成本，提高开发效率。

## 三 任务清单

### 1. 云服务器的选择（已完成）

通过租赁或购买云服务器能够免去本地服务器搭建和维护的成本，加快项目推进与展开。腾讯云提供了可以低价体验的云服务器套餐，其性能和资源对于项目初期的需要绰绰有余，并且能够找到较为完善的帮助文档，因此选择其作为搭建云服务器的平台，初步选择的服务器基本参数如下：

- 规格：S2.SMALL2
- CPU：Intel Xeon E5-2680 Broadwell（v4）@2.4GHz ×1
- 内存：2GB DDR4
- 公网带宽：1Mbps

- 内网带宽：1.5Gbps
- 体验价：10 元/月

之后，进行安装服务器操作系统（选择了 CentOS），以及 Python 的安装和环境配置。

## 2. web 应用框架的熟悉（基本完成）

通过阅读文档和相关教程，从 Flask 框架入手，熟悉 web 应用的编写和整体结构，并通过实际编写简单的 web 经典例程（如登录界面、邮件收发等），基本掌握 Flask 应用开发的流程，并配合 Gunicorn 尝试在云服务器上进行部署。

## 3. Python 程序编写（进行中）

使用 Python 的 socket 库完成 Python Socket 程序的编写，在云服务器端运行后，通过 PC 上的网络调试助手等工具验证与其进行 TCP 通信的可行性。

进一步可以编写简单的数据处理程序，在 PC 端发送模拟的波形数据到云服务器，验证完成数据处理并返回的流程。

## 4. 4G 模块通讯

使用 4G 模块和电脑通过 USB 转 TTL 模块连接后，从串口调试助手向 4G 模块发送指令和数据，代替直接通过网络调试助手的功能，此时，PC 模拟了数据采集终端产生数据的过程，来验证 4G 模块和云服务器通讯的可行性。

## 5. web 服务端开发

在 Flask 框架下编写 web 应用程序，初步实现数据访问，可视化显示等功能，进一步实现 web 应用向数据终端发送指令，动态地获取各个电力系统设备运行状态的功能。在此过程中也不断完善 SQLite 数据库结构，使其和服务端项目的各个部分紧密协作。

## 6. 进一步完善与尝试

完成了基本功能后，可以进一步完善服务器功能，如，增加数据处理部分增加更为复杂的处理功能，web 应用提供更为丰富的交互方式，云服务器和数据采集终端间更为可靠而高效的通讯，编写 61850 通讯协议接口，开发手机客户端等，进一步完善项目，并作出一些有益的尝试。

最后，感谢赵刚老师为我们提供这次锻炼和实践的机会，让我们能够从一个更为全面的角度来进行高压数字测量系统的设计，对电力系统的现实情况也有了更多的了解和认识。我们也会在本次课程设计中充分实践与尝试，锻炼自己的能力，做出成果，有所收获。