

哈爾濱工業大學

毕业设计（论文）中期报告

题 目：DNS 根服务数据分析系统设计与实现

专 业 信息安全

学 生 朱新锐

学 号 1180302003

指导教师 詹东阳

日 期 2022.03.24

哈尔滨工业大学教务处制

1. 论文工作是否按开题报告预定的内容及进度安排进行

1.1 论文工作开题预定的研究内容

1.1.1 论文研究背景

根域名服务器是 DNS 的分层树型结构中处于最顶层的服务器，是支持互联网运行的重要基础设施，离开了根域名服务器，DNS 将无法正常运行^[1]。一些国家、地区或组织机构引入了根服务器的根镜像服务器，来提供与根服务器相同的功能，而这些根服务器及与之相关的镜像服务器则分布在世界各地多个地区。中国大陆境内就架设了 22 台根镜像服务器。如果根域名服务器或其镜像的服务出现故障，可能会导致某些地区的 DNS 解析服务受到影响，乃至停止服务，那些依靠受影响的域名系统所支持的互联网应用和服务也将停止工作。而研究相关 DNS 根服务器、DNS 根镜像的安全问题则需要收集 DNS 根服务器与根镜像服务器的基本信息。

针对这个需求，本课题设计并实现一个根服务数据分析系统，用于收集分析由相关测量系统探测到的有关 DNS 根服务器或 DNS 根镜像的数据。分析这些数据有助于实时感知根服务器的运行状态，可以为类似于根服务器停止服务这样事件的应急响应提供支持。同时收集到的数据也可以用于评估测量节点所在地的 DNS 解析服务水平，为 DNS 根镜像服务器的部署提供参考。

1.1.2. 论文预定研究内容

在开题阶段，本论文工作的预定内容主要分为以下几个部分：

(1) DNS 根服务数据分析系统整体的设计与实现：

设计实现一个 DNS 根服务测量系统的数据，该系统能根据选定的分析方法实时分析测量数据，同时应当具备格式化存储测量数据与分析数据的能力。如果相协作的 DNS 探测系统开放相关权限的话，系统还应具备调度分配测量任务的能力。系统还能以图表的形式向操作人员展示数据。

分析系统具体由任务调度模块、分布式实时计算引擎、数据库、微服务模块和 Web 端组成，系统架构如图 1-1 所示。

任务调度模块负责接收由分布式 DNS 测量系统产生的实时测量数据，并将数据分发给计算引擎。该模块也负责计算引擎的负载平衡工作，降低对计算引擎的性能要求。

分布式实时计算引擎负责用于加工处理接收到的探测数据，并将处理结果交给数据库。

数据库模块存储接收到的原始数据与计算引擎输出的结果，并提供查询接口。

微服务模块位于数据库和用户的中间层，为用户提供查询、安排测量任务的等服务，同时实现对用户的访问控制，保护数据库的安全。

Web 端用于提供操作界面与图表展示。

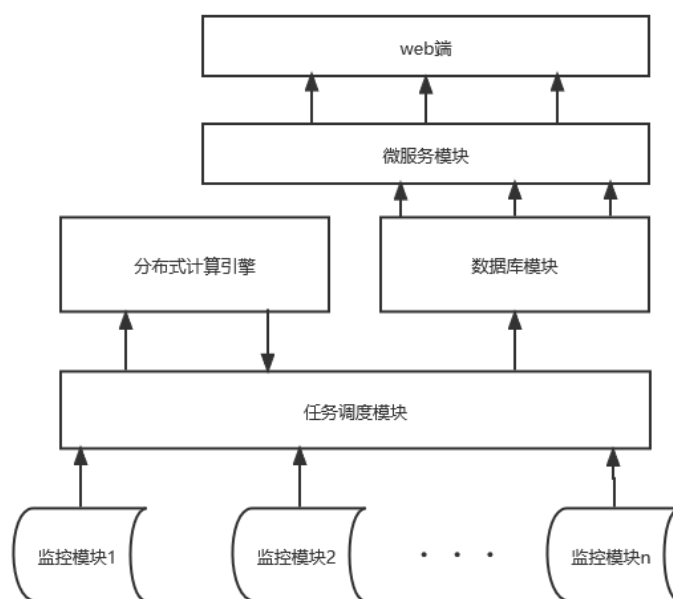


图 1-1 系统框架图

（2）数据分析方法的设计与实现

确定从原始数据中收集估计 DNS 服务器信息的方法。设计的方法应当能至少得到关于 DNS 服务器的延迟、可用性的信息，能进行简单的异常检测并响应。

1.1.3 论文预定进度安排

1. 2021 年 11 月：阅读文献，撰写开题报告。
2. 2021 年 12 月 - 2022 年 1 月：继续收集、分析资料，形成具体工作思路。
3. 2022 年 3 月 - 2022 年 4 月：初步完成分析系统的雏形，进行调试并发现不足之处。
4. 2022 年 5 月：改进系统，撰写毕业论文。

1.2 现阶段实际的论文工作及进度安排

目前论文工作按照开题的内容及进度安排推进。已经实现了 DNS 根服务数据分析系统中的任务调度模块、数据库模块、微服务模块与 web 端，与论文工作的进度安排与内容大体一致。

2. 已完成的研究工作及成果

研究工作主要为设计实现一个程序系统来收集根服务数据探测系统收集的数据并如图 2-1 所示进行可视化的展示，即实现了 DNS 根服务数据分析系统中的任务调度模块、数据库模块、微服务模块与 web 端。

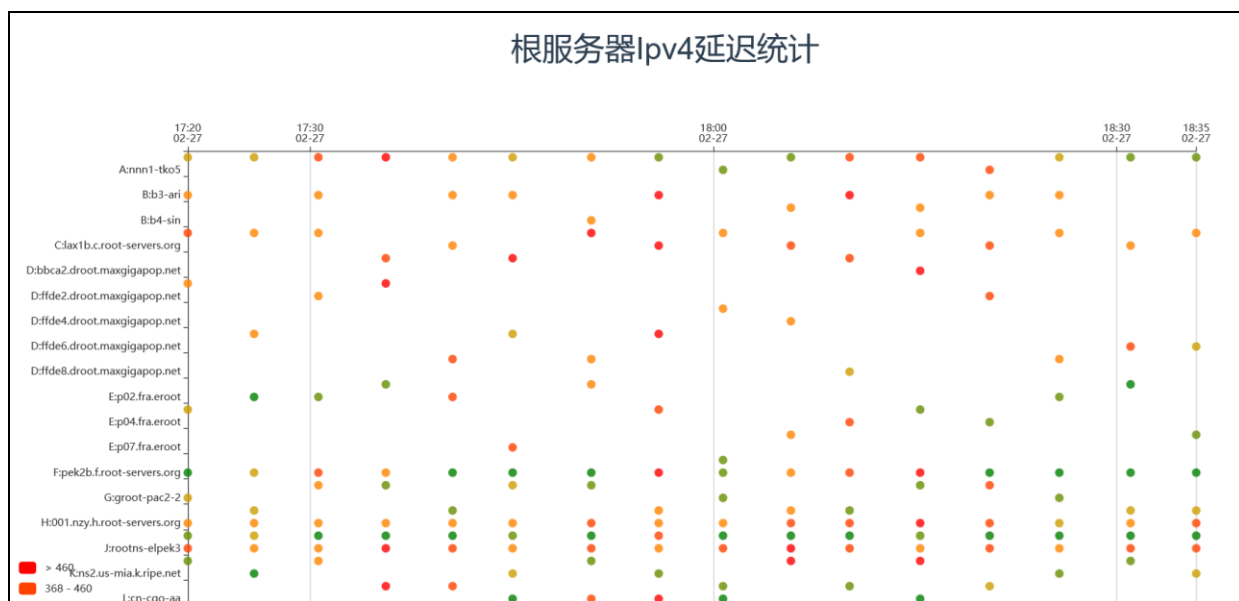


图 2-1 web 端数据展示示例

2.1 任务调度模块

本系统利用 Apache Kafka 实现任务调度模块。Apache Kafka 是一个高吞吐量的分布式消息系统，采用发布-订阅模式传递消息^[1]。系统运行一个 Kafka 服务器，根服务探测系统的探测节点会作为消息生产者(Producer)向 Kafka 服务器发送数据。由于目前还没有实现单独的计算引擎，实际处理探测数据的微服务模块会作为消息消费者(Consumer)从 Kafka 服务器拉取数据。

实现时系统遵循 Kafka 的默认负载均衡策略。Kafka 的分区器会先在生产者层面进行负载均衡，消息由分区器决定加入到哪个消息分区。在生产者不指定分区的前提下，消息如果 key 值非空，则基于 key 的 hash 值决定分区，key 值为空则向同一分区写入数据，直至达到吞吐的大小或时间限制，然后更换分区^[4]。在消费者层面 Kafka 通过协调消费者对对应消费的分区来进行再均衡。实现时遵循 Kafka 的默认分配策略即 range 策略，按照消费者总数和分区总数进行整除运算，获得一个跨度，然后将分区按照跨度进行平均分配^{[5][6]}。

2.2 数据库模块

由于预计实验的数据量大约在几百 MB 级别，数据库使用了 mysql 来实现。

数据库目前建立了三个表来存储来自探测节点的原始数据。分别如表 2-2、表 2-3 和表 2-4 所示，表 rsi_data 存储探测数据中除了路由信息之外的内容，rsi_route_data_ipv4 存储 Ipv4 路由数据，rsi_route_data_ipv6 存储 Ipv6 路由数据。

表 2-2 表 rsi_data

| 名称 | 描述 | 主键 |
|----------------------|-------------------|----|
| Timestamp | 时间戳 | 是 |
| Provider | 提供数据节点的标识 | 是 |
| Name | DNS 服务器的类型 | 是 |
| Identification | DNS 服务器的标识 | 是 |
| SourceIp_ipv4 | DNS 服务器的 IP v4 地址 | 否 |
| SourceIp_ipv6 | DNS 服务器的 IP v6 地址 | 否 |
| Latency_ipv4_udp | Ipv4、udp 协议下的延迟 | 否 |
| Latency_ipv4_tcp | Ipv4、tcp 协议下的延迟 | 否 |
| Latency_ipv6_udp | Ipv6、udp 协议下的延迟 | 否 |
| Latency_ipv6_tcp | Ipv6、tcp 协议下的延迟 | 否 |
| Status | DNS 解析状态 | 否 |
| Path_count_ipv4 | Ipv4 协议下的到服务器的跳步数 | 否 |
| Path_count_ipv6 | Ipv6 协议下的到服务器的跳步数 | 否 |
| refer_latency_ipv4_0 | Ipv4 协议下的三个参考延迟 | 否 |
| refer_latency_ipv4_1 | | 否 |
| refer_latency_ipv4_2 | | 否 |
| refer_latency_ipv6_0 | Ipv6 协议下的三个参考延迟 | 否 |
| refer_latency_ipv6_1 | | 否 |
| refer_latency_ipv6_2 | | 否 |
| correctness | 结果正确性 | 否 |
| publication_latency | 根区发布延迟 | 否 |

表 2-3 表 rsi_route_data_ipv4

| 名称 | 描述 | 主键 |
|----------------|------------|----|
| timestamp | 时间戳 | 是 |
| Provider | 提供数据节点的标识 | 是 |
| Goal | DNS 服务器的类型 | 是 |
| Identification | DNS 服务器的标识 | 是 |
| Index | 第几跳 | 是 |
| Address | 路由地址 | 否 |

表 2-4 表 rsi_route_data_ipv6

| 名称 | 描述 | 主键 |
|----------------|------------|----|
| timestamp | 时间戳 | 是 |
| Provider | 提供数据节点的标识 | 是 |
| Goal | DNS 服务器的类型 | 是 |
| Identification | DNS 服务器的标识 | 是 |
| Index | 第几跳 | 是 |
| Address | 路由地址 | 否 |

2.3 微服务模块

系统使用 spring-boot 框架实现微服务模块。这个部分作为客户与数据库的中间层，根据需求读写数据库数据。微服务包括 web 的后端，用于处理用户的查询请求；以及 Kafka Listener，用于拉取 Kafka 服务器中的数据，处理被收集的探测数据。

2.4 Web 端

Web 端使用 Vue 框架实现，用户可以用浏览器访问网址查看数据。目前实现了展示单个探测节点的探测数据。网页使用两个散点图分别展示 Ipv4 协议下的探测结果和 Ipv6 协议下的探测结果。图 2-1 展示了 Ipv4 协议下结果散点图的一部分。如图所示，每个点代表关于某时某台根服务器或某台根镜像的一条探测数据，点的横坐标表示时间，纵坐标代表服务器标识名，点的颜色反映 DNS 延迟的大小，颜色越绿代表延迟越低，越红则代表延迟越高。网页会定期向服务器请求最新数据。

如图 2-5 所示，网页还提供了一个表格来展示包括路由信息在内详细数据，以及一个表单供用户限定数据搜索的范围，包括限定时间段、探测节点与服务器的类别。



图 2-5 网页的详细数据展示

3. 后期拟完成的研究工作及进度安排

- (1) 4 月：在微服务模块添加周期生成报告的功能，报告将依据 RSSAC Advisory on Metrics for the DNS Root Servers and the Root Server System 一文的要求，根据收集到的探测数据生成报告，评估根服务器或其镜像的可用性等信息，web 端相应提供展示与下载功能；为系统添加调度探测节点的功能，允许授权的用户调整部分探测节点的探测任务，并在 web 端添加对应的界面；改进 web 端数据展示效果，美化界面。
- (2) 5 月：部署根服务数据分析系统并运行，收集实验数据，根据实验数据对系统进行改进，同时开始毕业论文的撰写。
- (3) 6 月：完成毕业论文，进行终期答辩。

4. 存在的问题与困难

(1) 目前系统仅能被动地接收探测节点发送的数据，不能主动地调整探测节点的探测任务，也缺少管理探测节点的能力，需要查阅资料，完善涉及分布式的设计，实现调度调整探测节点的功能。

(2) 数据库目前可靠性不足，需要定期备份等措施来提高可靠性。

(3) 没有实现计算引擎模块，系统难以处理复杂的分析任务。

5. 论文按时完成的可能性

按照进度安排，尝试解决现阶段出现的问题并逐步完成后续工作。就实现添加系统主动地调整探测节点的探测任务、添加管理探测节点的功能而言，预计使用 **Kafka** 可以满足通信需求。就改进数据库可靠性而言，**mysql** 备份的具体实现方案预计也可以在查阅相关资料后确定。就实现计算引擎而言，独立出来的计算模块预计可以采用流处理相关技术实现。论文可以按时完成。

6. 参考文献

- [1] 于世梁. 浅谈根域名服务器与国家网络信息安全[N].江西行政学院报. 2013 年 4 月, 第 15 卷第 2 期
- [2] Root Server Technical Operations Association [EB /OL] . <https://root-servers.org/>, 2021-11-25
- [3] 吴璨,王小宁,肖海力,曹荣强,赵一宁,迟学斌.分布式消息系统研究综述[J].计算机科学,2019,46(S1):1-5+34.
- [4] Apache Kafka [EB /OL] . https://kafka.apache.org/documentation/#producerconfigs_partitioner.class, 2022.3.21
- [5] Apache Kafka [EB /OL] . https://kafka.apache.org/documentation/#consumerconfigs_partition.assignment.strategy, 2022.3.21
- [6] RangeAssignor (kafka 2.4.0 API) [EB /OL] <https://kafka.apache.org/24/javadoc/org/apache/kafka/clients/consumer/RangeAssignor.html>. , 2022.3.21