

毕设中期报告

大数据（信息与计算科学（数据科学与大数据方向））马一清 2018.4.24

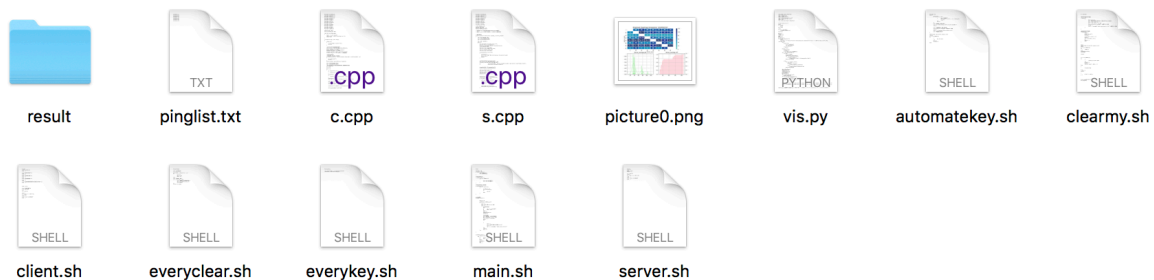
详情参考<https://github.com/miqianmimi/pingmesh-graduate-project-2018>

总体进展：

截止到4/24/2018：已完成pingmesh系统的最初版本demo.

pingmesh系统广泛得可以划分成三个部分

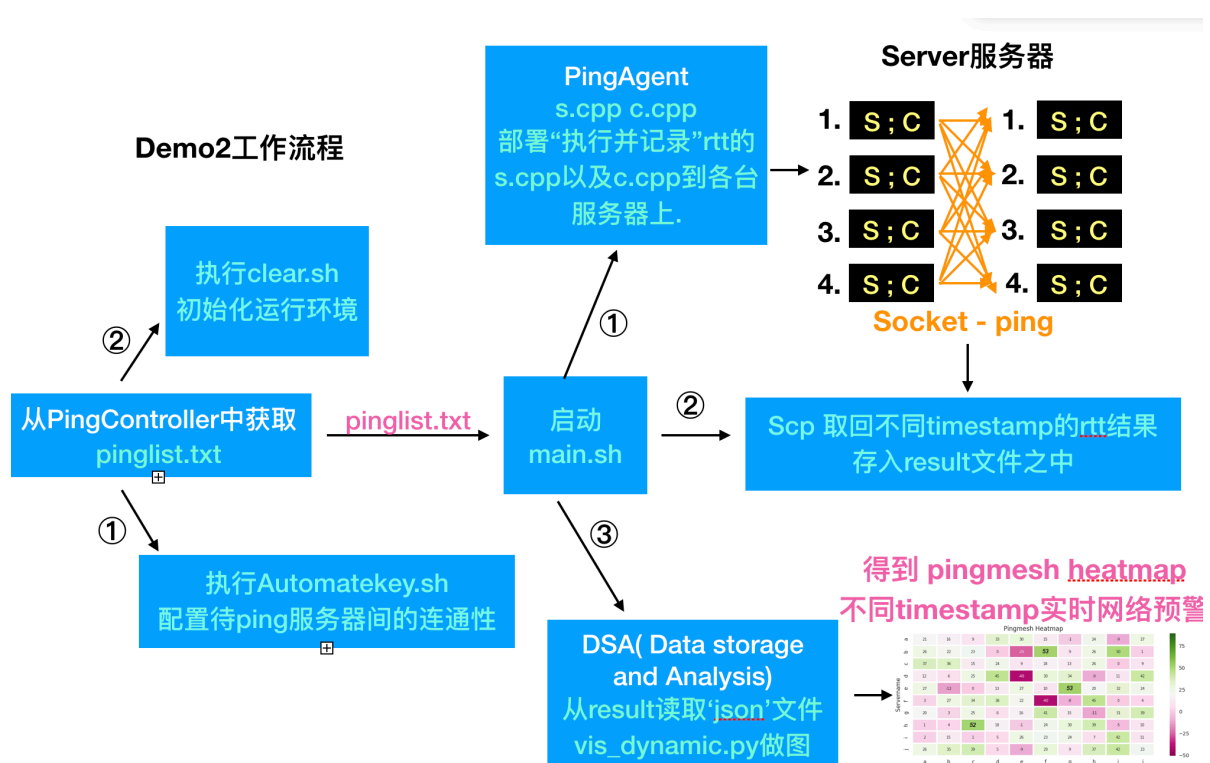
“pingagent”；“pingcontroller”；“DSA存储+可视化”；



pingagent部分由s.cpp；c.cpp 构成，他们由C++语言完成，使用socket套接字编程和fork实现同一服务器端开启多个端口监听客户端，并收集JSON文档，写入文档。

Pingcontroller部分 由 pinglist.txt，main.sh，clearmy.sh, automatekey.sh构成，他们由shell脚本语言完成，clearmy.sh和automatekey.sh分别负责清理环境和生成公钥简化服务器ssh通讯。main.sh 根据pinglist.txt 把pingagent发送到服务器上，开始Ping,得到数据后，存储起来。

DSA部分 由vis.py 构成，主要由Python完成，主要把存储起来的数据可视化，得到pingmesh热力图。



每日DAILY进展：

4/10

- | |
|--|
| 0.定义数据格式Log |
| 1.实现python pingmesh简单10v10可视化——+latency cdf图 |
| 2.实现Python pingmesh简单40V40可视化 |
| 3.构造自动生成数据程序 |

4/11

- | |
|--------------------|
| 0.第一个C++程序 |
| 1.实现socket 简易版通讯 |
| 2.实现socket 人机交互版通讯 |

4/12

- | |
|------------------------------------|
| 1.改成:server ping 多次 client 多次 fork |
| 2.改成文件储存版本，并且存到server端 |
| 3.改成时间3S一发送 |

4/13

- | |
|-----------------------|
| 1.实现Server加个time的PING |
| 2.实现两台服务器之间的ping |
| 3.实现自动计时 |
| 4.实现计时并且得到log数据 |

4/14

- | |
|--|
| 1.完成log输出
[Timestamp, SrcIP, SrcPort, DstIP, DstPort, Protocol, ProbingType, MsgLen, RTT, ErrCode], ...}} |
| 2.定三台服务器100,101,102，获取他们的数据. |
| 3.完成多个shell脚本/python模拟脚本，记录nc tcp的时间，得到baseline |
| 4.参考goaccess做log文档直接分析可视化监控。 |

4/16

- | |
|------------|
| 1.json格式输出 |
|------------|

4/17

1. 读出文件到python;用json画图
2. 一体化操作
1. 查后面5个哪里出来的
2. 写了clear程序，调出了不对等的bug
3. 初始demo,shell读数据C++PING,python画图一体化，以两个server为例

4/19

Pinglist main10*10.sh clear10*10.sh 配密钥10*10.sh
添加clearmy.sh，实现pingmesh之后复原工作
添加automatickey.sh，实现pingmesh自动配网关到服务器之间密钥

4/21

第二版demo,shell读数据，clean，key作用，C++PING,python画图一体化，以两个server为例
后台两次&并行，使得同时获得同一时刻所有数据
根据pinglist中server个数，自动实现获取服务器两两ping的数据
第二版demo实现根据pinglist，获得任意n个server Ping的结果。

4/23

8*8 服务器server client；4个Timestamp图
Python采用automatic动图；能够生成不同timestamp下的图

4/24

1.expect 和 spawn和key-generate 免除钥匙自动登录
2.&实现后台操作的shell
3.写一个shell完成自动化操作

存在的问题及下阶段工作打算：

【general 1】

可采用SQLite; Kafka(消息队列) (data bus) 存储数据，方便对数据进行进一步分析，并且实现动态写入，动态读取画图。

【general 2】

为了实现更快，更佳的同时起多个监听器，应由epoll代替fork。实现2500台服务器互相ping的情况下，CPU使用率0.25%，内存在45MB。

【general 3】

python 实现分析图形数据显得比较初级，欲将首先实现echart进行动态数据分析，在实现网页端动态数据显示，结合javascript,d3,以及grafana工具

【general 4】

目前，在小规模集群上，PingController并不需要执行算法来决定Pinglist挑选，当Pingmesh部署到云服务器端上时，服务器数量上涨到10000数量级，需要决定根据服务器拓扑结构决定pinglist的挑选。并且画图的程序也需要根据tor,spine来滚轮调整视野。

【general 5】 除了测试简单的ping，可测试RDMA的ping的延迟数据。

【detail1】

实现并行的ping的程序现在写在bash里，应修改到C++Server之内

【detail2】

现行保存的json格式不太适合server变多时的量化扩展。有待优化

