论文中期答辩：数据中心网络监控系统pingmesh

https://github.com/miqianmimi/pingmesh-graduate-project-2018

Pingmesh: A Large-Scale System for Data Center Network Latency Measurement and Analysis

Chuanxiong Guo, Lihua Yuan, Dong Xiang, Yingnong Dang, Ray Huang, Dave Maltz,
Zhaoyi Liu, Vin Wang, Bin Pang, Hua Chen, Zhi-Wei Lin, Varugis Kurien†
Microsoft, †Midfin Systems

SIGCOMM 2015

Yiqing Ma
26/04/2018

Why pingmesh

历史：Pingmesh是微软在2015年提出的，
其架构模式在那之前已经在微软的数据中心运行了超过4年

解决问题：

- 定位服务器的延迟是否因为网络
- 提供并且跟踪当前网络服务水平(SLA)
- 自动排除障碍

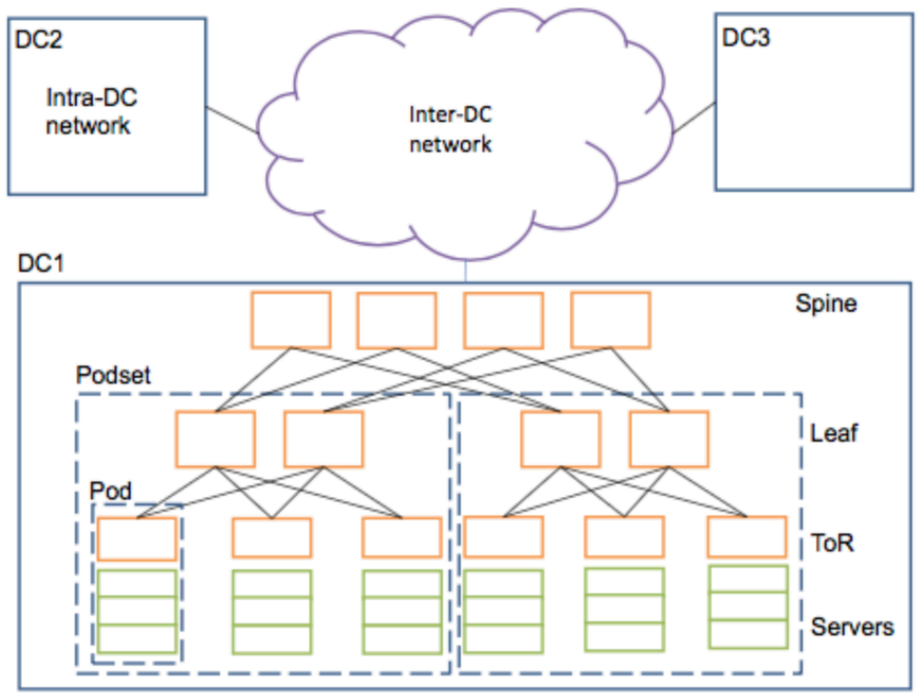
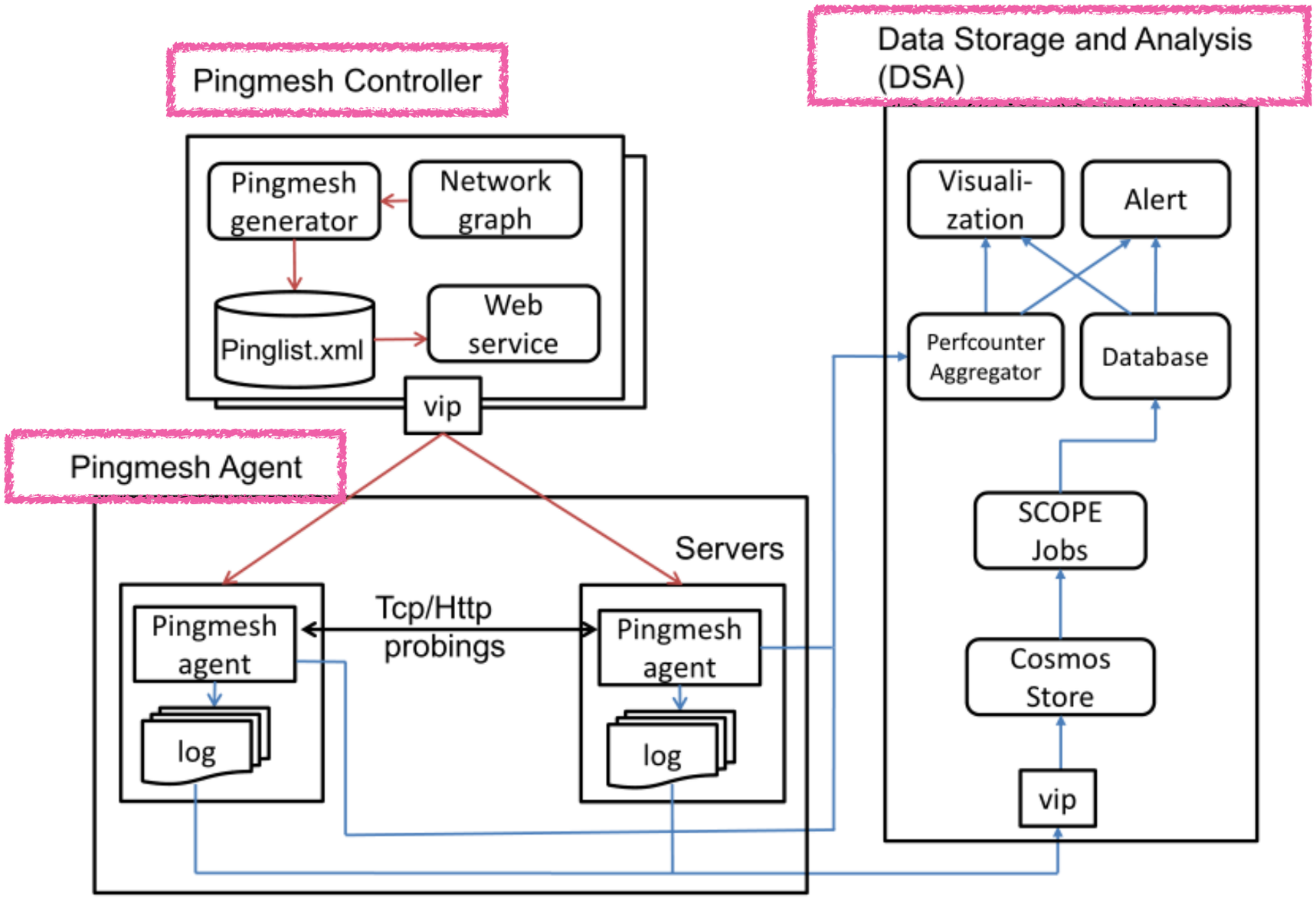


Figure 1: Data center network structure.

微软数据中心采用CLOS架构。

拥有的规模如下：
服务器规模在10万级别，
交换机规模为万，
服务器上联万兆.

How pingmesh



Demo2模块组件

PingAgent
1.s.cpp 2.c.cpp

PingController
1.main.sh
2.clear.sh
3.Automatekey.sh
4.pinglist.txt

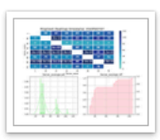
DSA(Data storage and Analysis)
vis.py
vis_dynamic.py


result

pinglist.txt

c.cpp

s.cpp


picture0.png

vis.py

automatekey.sh

clearmy.sh

client.sh

everyclear.sh

everykey.sh

main.sh

server.sh

Demo2工作流程

PingAgent
s.cpp c.cpp
部署“执行并记录”rtt的
s.cpp以及c.cpp到各台
服务器上.

Server服务器

1. S ; C　1. S ; C
2. S ; C　2. S ; C
3. S ; C　3. S ; C
4. S ; C　4. S ; C

Socket - ping

执行clear.sh
初始化运行环境

②

①

Operator

从PingController中获取
pinglist.txt

Operator

pinglist.txt

启动
main.sh

②

Scp 取回不同timestamp的rtt结果
存入result文件之中

Host

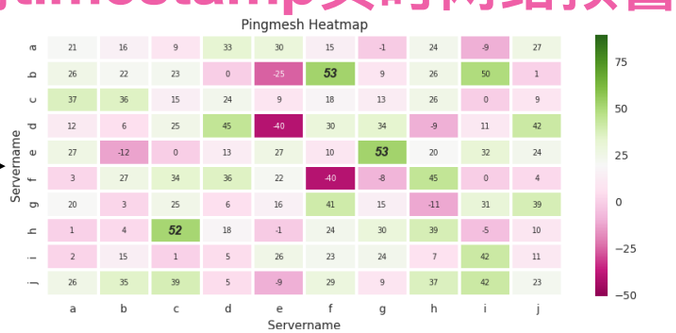
①

执行Automatekey.sh
配置待ping服务器间的连通性

③

DSA(Data storage
and Analysis)
从result读取json文件
vis_dynamic.py做图

得到 pingmesh heatmap
不同timestamp实时网络预警



流程展示：

Demo2到Demo3该如何改进

⑤ 【general 5】

除了测试简单的ping，可测试RDMA的ping的延迟数据.

执行clear.sh
初始化运行环境

从PingController中获取

pinglist.txt

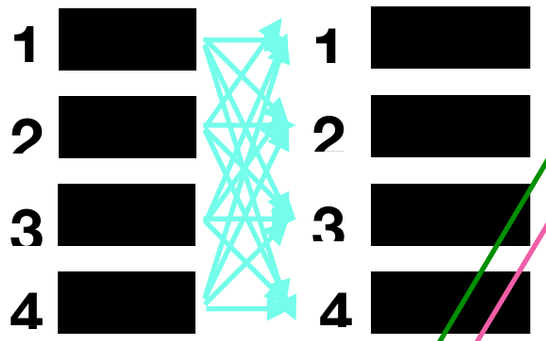
启动 main.sh

① 【detail1】

实现并行的ping的程序现在写在bash里，应修改到C++Server之内

PingAgent
s.cpp c.cpp
部署“执行并记录”rtt的s.cpp以及c.cpp到各台服务器上.

Server服务器



Socket - ping： fork

Scp 取回不同timestamp的rtt结果

① 【general 1】

可采用SQLite; Kafka(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。

② 【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

② 【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。

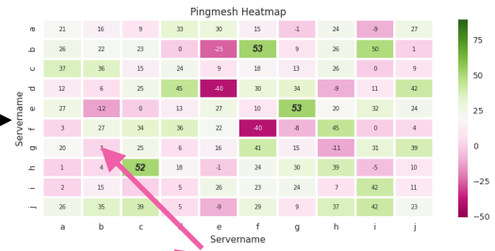
执行Automatekey.sh
配置待ping服务器间的连通

④ ★ 【general 4】

目前，在小规模集群上，Ping Controller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。

DSA(Data storage and Analysis)
从result读取‘json’文件
vis_dynamic.py做图

得到timestamp对应的pingmesh：heatmap



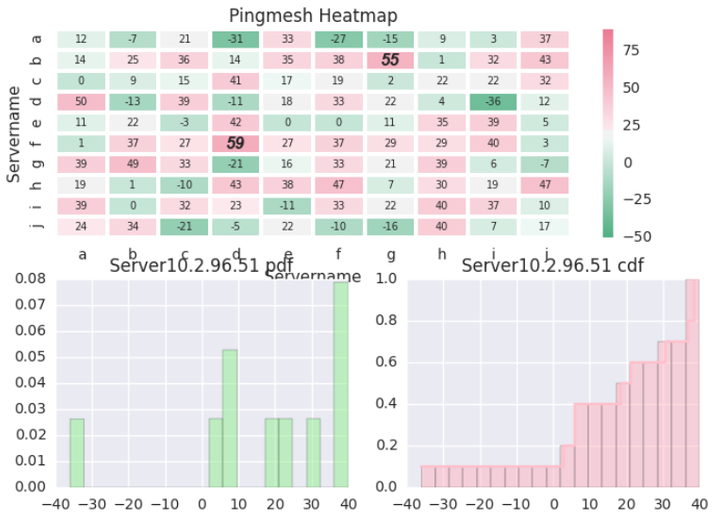
③

【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

已有可视化成效图：

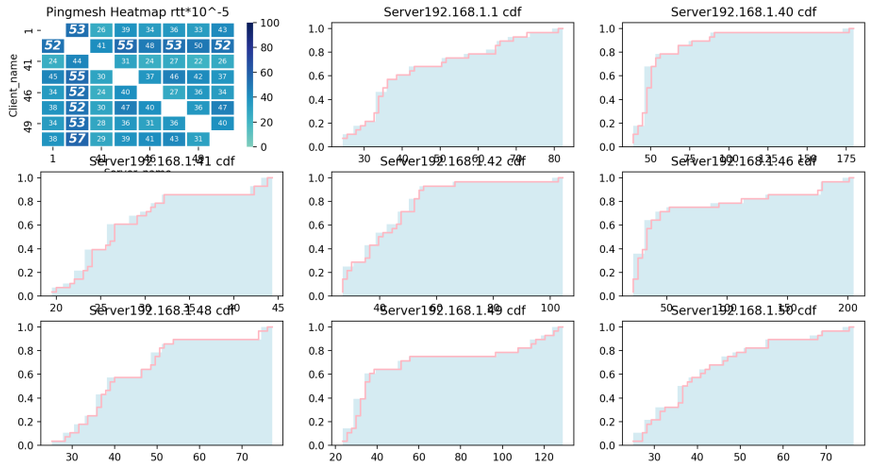
Demo1



采用模拟生成数据

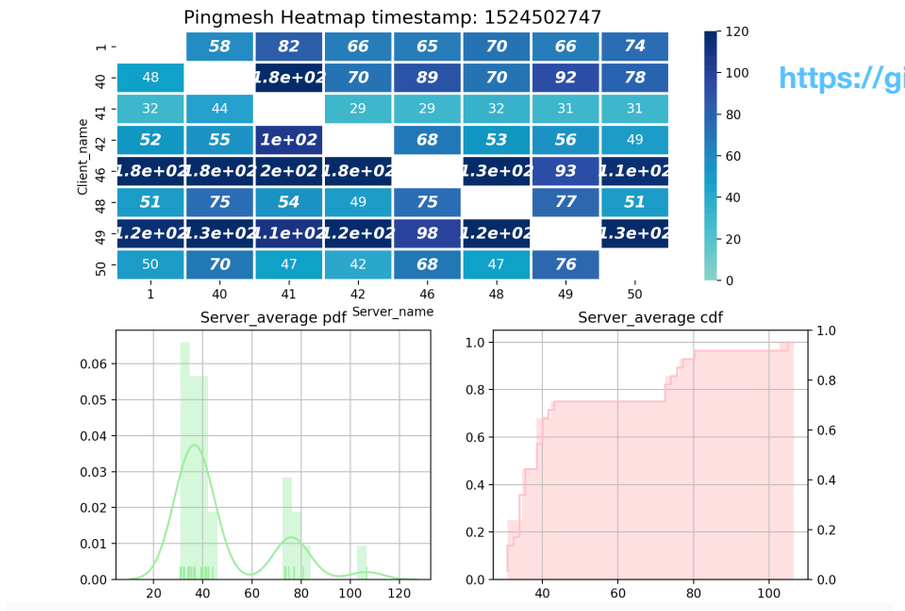
Demo2

Cdf of each server's latency



采用真实ping数据

Heat map of ping rtt



Cdf of average latency

数据来源：

HKUST Cluster-Sing.md

143.89.191.114

https://github.com/HKUST-SING/Equipment-SINGLab

8台服务器

192.168.1.1

192.168.1.40

192.168.1.41

192.168.1.42

192.168.1.46

192.168.1.48

192.168.1.49

192.168.1.50

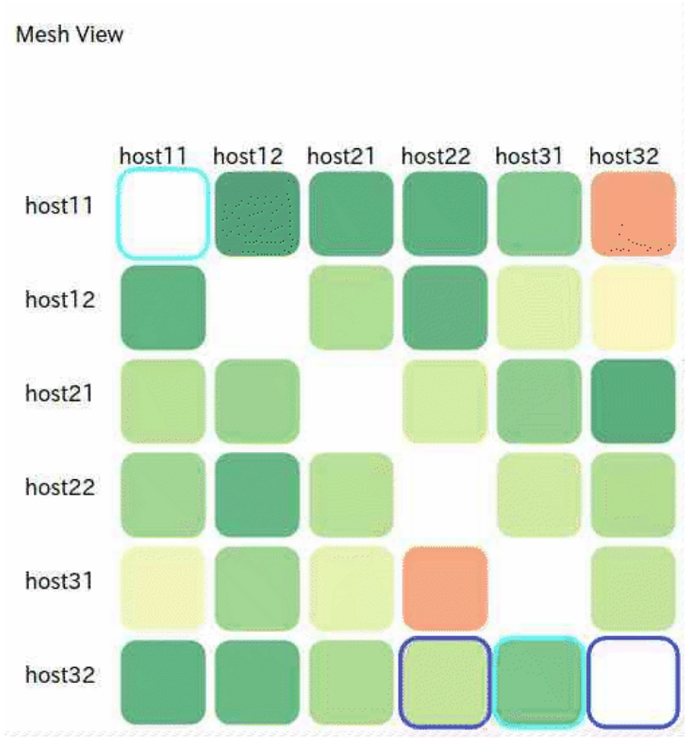
预期可视化成效图

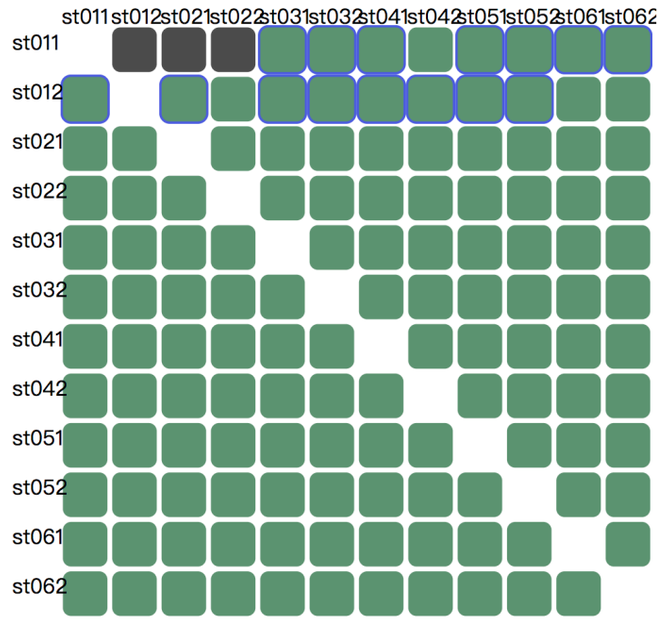
grafana dashboard ——用于latency cdf 实时

Js d3 网页前端——用于 heatmap 实时

结合两者

最终effect





每日进展：
详情参考我的github
https://github.com/miqianmimi/pingmesh-graduate-project-2018

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.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

4/13

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。

4/14

1.json格式输出

4/16

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

4/17

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

4/19

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

4/21

Pinglist main10*10.sh clear10*10.sh 配秘钥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秘钥

4/23

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。

4/24

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