



中國計算機學會
CHINA COMPUTER FEDERATION



Nightingale



GitLink
— 确实 · 开源 —

第二届CCF·夜莺开发者创新论坛

中国北京 2024.7.26

主办方: 中国计算机学会 | 承办方: CCF开源发展委员会、夜莺项目开源社区



中國計算機學會
CHINA COMPUTER FEDERATION



Nightingale



GitLink
— 确实 · 开源 —

网络设备监控中的陷阱及解法

夜莺SNMP监控最佳实践

中国北京 2024.7.26

主办方: 中国计算机学会 | 承办方: CCF开源发展委员会、夜莺项目开源社区

大纲

第二届CCF·夜莺开发者创新论坛

1

网络设备可观测性

2

网络设备指标监控

3

网络设备 SNMP 采集陷阱

4

Catgraf 采集 SNMP 最佳实践

中国北京 2024.7.26



中國計算機學會
CHINA COMPUTER FEDERATION



#01

网络设备可观测性

指标 (Metrics) 和日志 (Log)

网络设备指标

- 通过 SNMP 和遥测 (Telemetry) 收集
- 反映设备性能、状态和资源使用情况
- 常见指标: CPU 使用率、内存使用率、网络接口流量、错误率等

网络设备日志

- 记录设备运行过程中的事件和操作
- 帮助检测和诊断问题、追踪设备活动
- 常通过 Syslog 协议进行采集和传输

#02

网络设备指标监控

背景和架构

中国北京 2024.7.26



中國計算機學會
CHINA COMPUTER FEDERATION

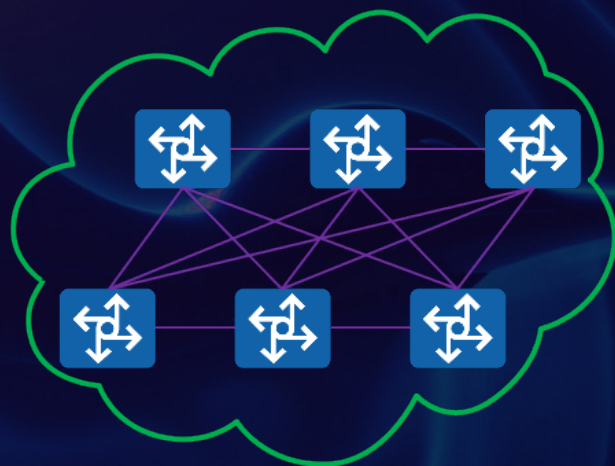


网络设备监控背景

- 网络设备领域还是以 Zabbix 为主，Prometheus 体系弱势
- 内部统一监控告警平台都是基于 Prometheus 体系
- 网络设备厂商封闭，还不够拥抱开源

网络设备指标采集协议

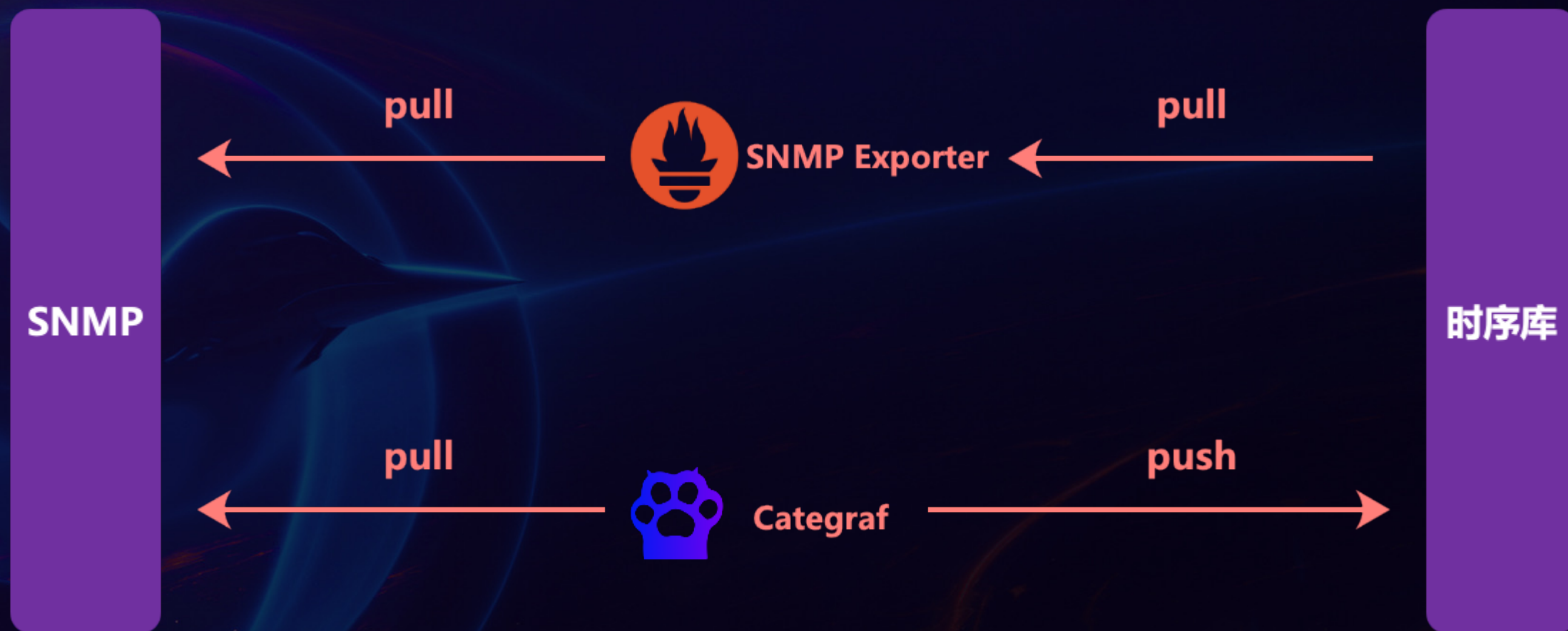
网络数据源



指标采集器



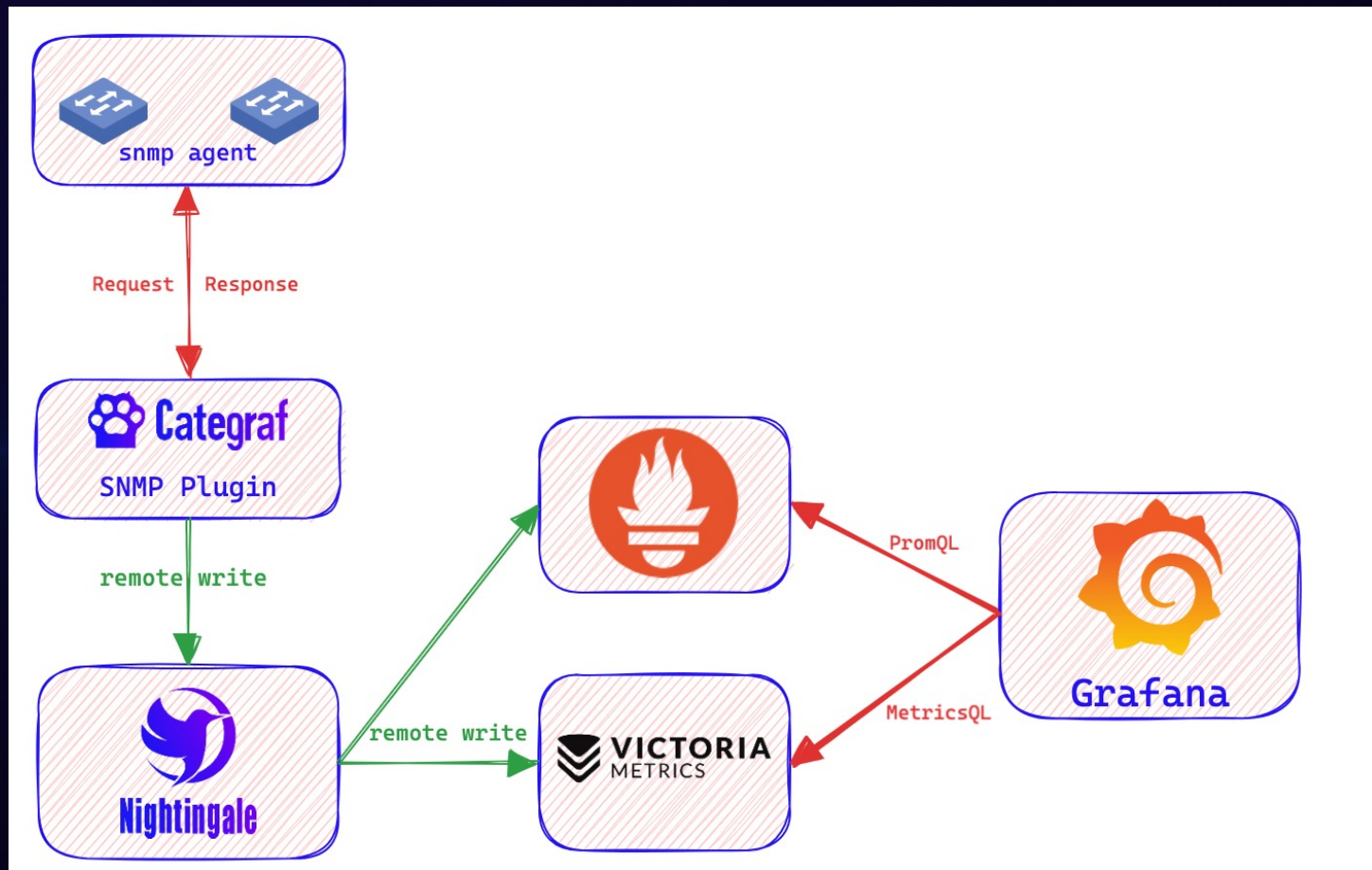
网络设备指标采集器



网络设备监控架构

- 采集器：Categraf 中的 SNMP 插件
- 远程写：夜莺
- 时序库：VictoriaMetrics
- 可视化：Grafana

夜莺最新版本可直接集成 Grafana，实现了直接在夜莺中管理。



#03

网络设备采集陷阱

SNMP设备

网络设备可视化



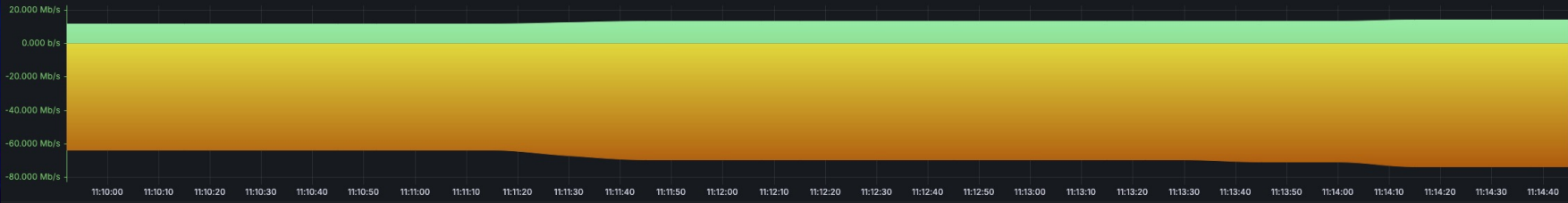
交换机汇总【总数: 18】

设备IP	设备名	启动(天)	CPU使用率	内存使用率	PING连通性
172.17.0.1	HZZB-NL-AG-Master	127	4%	24%	正常
172.17.0.2	HZZB-NL-AG-Master	178	8%	49%	正常
172.17.0.3	HZZB-NL-AG-Master	17.7	8%	58%	正常
172.17.0.4	HZZB-NL-AG-Master	23.8	11%	11%	正常
172.17.0.5	HZZB-NL-AG-Master	23.6	11%	50%	正常
172.17.0.6	HZZB-NL-AG-Master	214	11%	58%	正常

接口信息汇总

接口名称	上行速率	下行速率	接口速率	CRC错包	接口状态
Eth-Trunk1	14.13 Mb/s	73.66 Mb/s	40 Gb/s	非物理接口	UP
XGigabitEthernet1/0/4	4.16 Mb/s	45.28 Mb/s	10 Gb/s	0	UP
XGigabitEthernet0/0/4	2.56 Mb/s	22.46 Mb/s	10 Gb/s	0	UP
Eth-Trunk2	67.05 Mb/s	12.96 Mb/s	4 Gb/s	非物理接口	UP
XGigabitEthernet1/0/3	4.64 Mb/s	5.78 Mb/s	10 Gb/s	0	UP
GigabitEthernet1/0/20	20.38 Mb/s	4.62 Mb/s	1 Gb/s	0	UP
GigabitEthernet1/0/19	25.11 Mb/s	3.54 Mb/s	1 Gb/s	0	UP

指定接口: (Eth-Trunk1) 上行/下行



Name	Min	Max	Last *	Mean
上行: Eth-Trunk1	11.677 Mb/s	14.127 Mb/s	14.127 Mb/s	12.861 Mb/s
下行: Eth-Trunk1	-73.659 Mb/s	-63.756 Mb/s	-73.659 Mb/s	-68.247 Mb/s

网络设备可视化

光模块信息

光模块状态 ⓘ

接口 ⌵	传输距离	波长	光模块模式	接收功率	发送功率	接收功率(高预警)	接收功率(低预警)	发送功率(高预警)	发送功率(低预警)
XGigabitEthernet0/0/1 ⓘ ⓘ	1400(9um)	1310	singleMode	-1.28 dBm	-2.79 dBm	0.50 dBm	-14.39 dBm	0.50 dBm	-8.19 dBm
XGigabitEthernet0/0/2	1400(9um)	1310	singleMode	-2.24 dBm	-2.15 dBm	0.50 dBm	-14.39 dBm	0.50 dBm	-8.19 dBm
XGigabitEthernet0/0/3	1400(9um)	1310	singleMode	-4.61 dBm	-3.72 dBm	0.50 dBm	-14.39 dBm	0.50 dBm	-8.19 dBm
XGigabitEthernet0/0/4	1400(9um)	1310	singleMode	-4.65 dBm	-2.99 dBm	0.50 dBm	-14.39 dBm	0.50 dBm	-8.19 dBm
XGigabitEthernet1/0/1	1400(9um)	1310	singleMode	-1.73 dBm	-1.88 dBm	0.50 dBm	-14.39 dBm	0.50 dBm	-8.19 dBm
XGigabitEthernet1/0/2	1400(9um)	1310	singleMode	-2.73 dBm	-2.34 dBm	0.50 dBm	-14.39 dBm	0.50 dBm	-8.19 dBm
XGigabitEthernet1/0/3	1400(9um)	1310	singleMode	-4.37 dBm	-3.78 dBm	0.50 dBm	-14.39 dBm	0.50 dBm	-8.19 dBm
XGigabitEthernet1/0/4	1400(9um)	1310	singleMode	-5.73 dBm	-3.57 dBm	0.50 dBm	-14.39 dBm	0.50 dBm	-8.19 dBm

LLDP信息

交换机LLDP信息 ⓘ

本地设备IP	本地端口名称	邻居端口名称	邻居设备名称	邻居端口描述	本地端口描述	本地设备名称
172.18.112.2	GigabitEthernet0/0/19	GigabitEthernet1/0/20	HZ-ZB...	...	to-18...	HZZB-NL...
172.18.112.2	GigabitEthernet0/0/20	GigabitEthernet0/0/20	HZ-ZB...	...	to-18...	HZZB-NL...
172.18.112.2	GigabitEthernet0/0/21	GigabitEthernet0/0/47	HZ-ZB...	...	to-18...	HZZB-NL...
172.18.112.2	GigabitEthernet0/0/23	GigabitEthernet0/0/23	HZ-ZB...	...	to-18...	HZZB-NL...
172.18.112.2	GigabitEthernet0/0/28	GigabitEthernet1/0/28	HZZB-...	GigabitEthernet1/0/28	GigabitEthernet0/0/28	HZZB-NL...

常见陷阱01

- MIB文件解析程序模式选择
 - gosmi + path
 - netsnmp + PATH

常见陷阱01

- path: mib 文件路径
- transtator : gosmi
- 位置：内置库，系统无需做其他操作
- mib：放入全部的mib即可

path + gosmi 模式强烈推荐

```
[[instances]]
agents = [
    "udp://172.16.10.10",
    "udp://172.16.10.11"
]

...

path = ["/opt/categraf/mibs/huawei", "/opt/categraf/mibs/public"]
translator = "gosmi"
# translator = "netsnmp"

...
```

常见陷阱01

- PATH: mib 文件路径设置为环境变量
- transtator : netsnmp
- 位置：外置库，系统需要安装 netsnmp
- mib：放入全部的mib即可

PATH + netsnmp 模式兼容性更强

```
[[instances]]
agents = [
    "udp://172.16.10.10",
    "udp://172.16.10.11"
]

...

# path = ["/opt/categraf/mibs/huawei", "/opt/categraf/mibs/public"]
# translator = "gosmi"
translator = "netsnmp"

...
```

常见陷阱01

```
export MIBDIRS="/opt/categraf/mibs/huawei:/opt/categraf/mibs/mibs/public"
```

```
# 执行categraf命令测试采集 snmp 查看异常的输出  
/opt/categraf/categraf --test --inputs snmp
```

```
# 如果在启动 Categraf 时候需要指定 MIB 文件读取目录，就在 categraf.service 文件中指定该环境变量  
Environment="MIBDIRS=/opt/categraf/mibs/huawei:/opt/categraf/mibs/mibs/public"
```

常见陷阱02

- MIB 引入依赖文件缺失
- MIB 文件格式错误

```
root@alert:/opt/categraf# /opt/categraf/categraf --test --inputs snmp
2024/07/17 17:13:12 main.go:149: I! runner.binarydir: /opt/categraf
2024/07/17 17:13:12 main.go:150: I! runner.hostname: alert
2024/07/17 17:13:12 main.go:151: I! runner.fd_limits: (soft=1048576, hard=1048576)
2024/07/17 17:13:12 main.go:152: I! runner.vm_limits: (soft=unlimited, hard=unlimited)
2024/07/17 17:13:12 provider_manager.go:60: I! use input provider: [local]
2024/07/17 17:13:12 prometheus_agent.go:19: I! prometheus scraping disabled!
2024/07/17 17:13:12 ibex_agent.go:19: I! ibex agent disabled!
2024/07/17 17:13:12 agent.go:38: I! agent starting
Parse module: /opt/categraf/mibs/h3c/public/ipsec-ikeaction.mib:36:18: unexpected "20060905'" (expected <extutctime> ...)
2024/07/17 17:13:25 translate.go:84: W! Couldn't load module ipsec-ikeaction.mib: Could not load module at ipsec-ikeaction.mib
Parse module: /opt/categraf/mibs/h3c/public/rfc1155.smi:20:9: unexpected "SMI" (expected "END")
2024/07/17 17:13:25 translate.go:84: W! Couldn't load module rfc1155.smi: Could not load module at rfc1155.smi
Parse module: /opt/categraf/mibs/h3c/public/rfc1212.smi:3:5: unexpected "SMI" (expected "END")
2024/07/17 17:13:25 translate.go:84: W! Couldn't load module rfc1212.smi: Could not load module at rfc1212.smi
2024/07/17 17:13:46 metrics_agent.go:317: I! input: local.snmp started
2024/07/17 17:13:46 agent.go:46: I! [*agent.MetricsAgent] started
2024/07/17 17:13:46 agent.go:49: I! agent started
```

常见陷阱02

- 建立对应文件夹管理不同品牌的MIB文件，私有MIB库和公有MIB库分开管理

```
root@alert:/opt/categraf/mibs# tree -L 1
```

```
.  
├── dell  
├── fortinet  
├── h3c  
├── huawei  
├── public  
└── sangfor
```

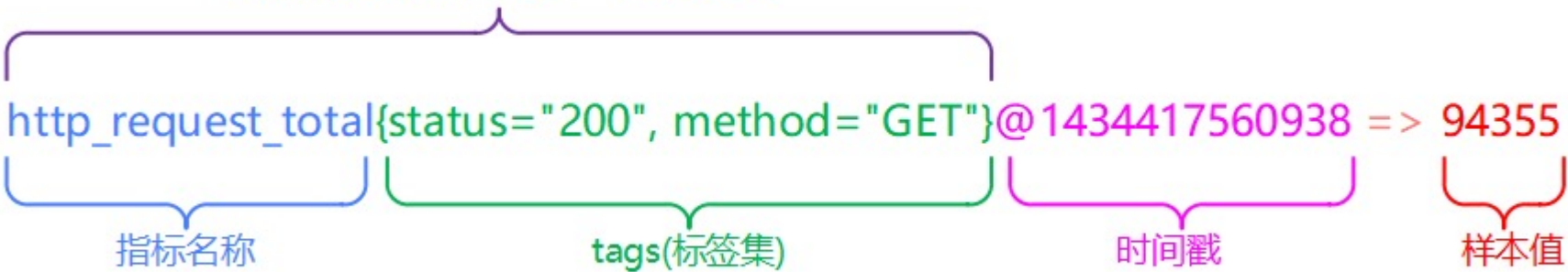
```
6 directories, 0 files
```

```
interval = 15  
  
[[instances]]  
agents = [  
    "udp://192.168.0.252"  
]  
  
...  
  
path = ["/opt/categraf/mibs/huawei", "/opt/categraf/mibs/public"]  
translator = "gosmi"  
  
...
```

常见陷阱03

- 指标数据只采集数值，不采集字符串，标签维持稳态结构
- 合理规划标签和指标，避免浮点数陷阱

指标名称和标签集构建一个样本指标



常见陷阱03

- Gauge、Counter、Integer 可直接作为指标值和标签采集
- String 只可作为标签采集

```
# 系统启动时间
[[instances.field]]
oid = ".1.3.6.1.2.1.1.3.0"
name = "uptime"

# 系统名称
[[instances.field]]
oid = ".1.3.6.1.2.1.1.5.0"
name = "device_name"
is_tag = true

...

# 实体名称
[[instances.table.field]]
oid = ".1.3.6.1.2.1.47.1.1.1.7"
name = "ent_name"
is_tag = true

# 实体内存实时利用率
[[instances.table.field]]
oid = ".1.3.6.1.4.1.25506.2.6.1.1.1.8"
name = "mem_usage"
```

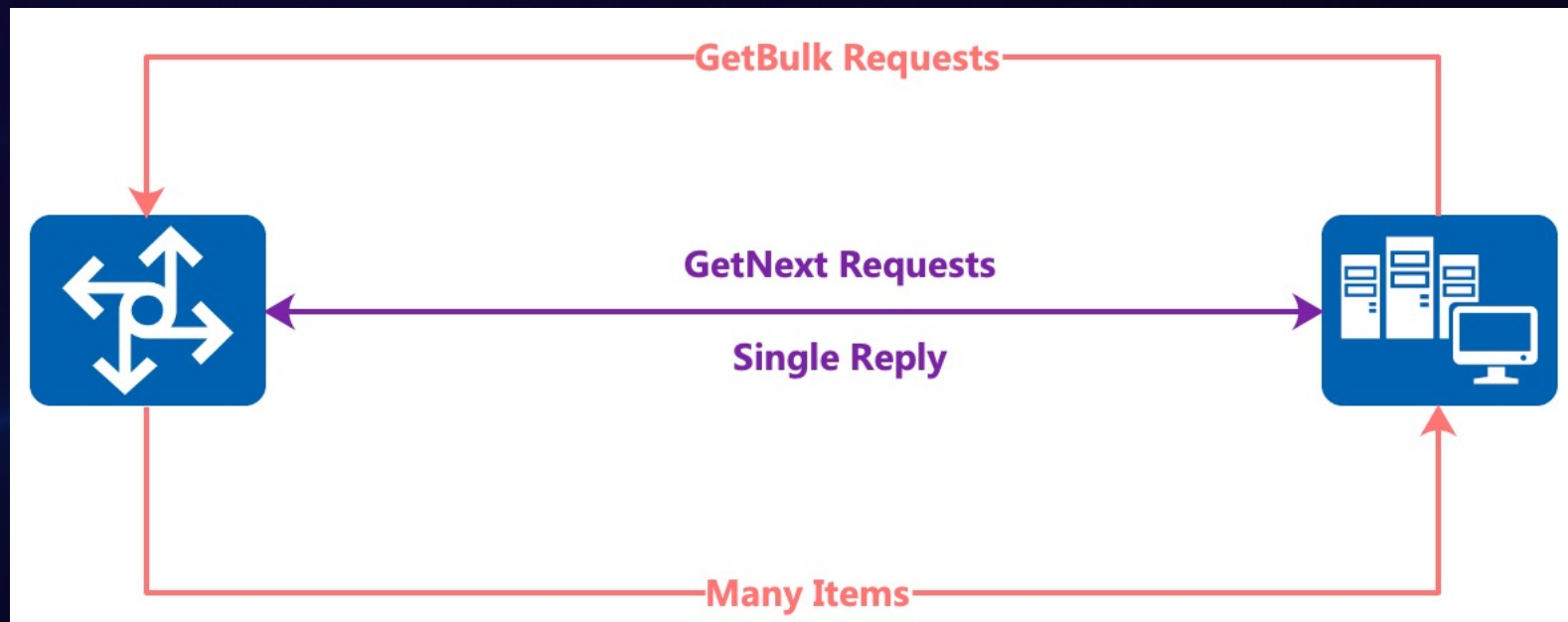
常见陷阱04

- performing bulk walk
- request timeout after 3 retries

```
instances.go:186: agent udp://172.16.x.x ins: gathering table system error:  
performing bulk walk for field metrics_name(.1.3.6.1.2.1.47.1.1.1.1.2):  
request timeout after 3 retries
```

常见陷阱04

- GetNext 一次请求
单个回复
- GetBulk 一次请求批
量回复



常见陷阱04

特性	GetNext	GetBulk
效率	较低，每次一个对象	较高，一次多个对象
请求次数	多次请求	一次请求
支持版本	SNMPv1、SNMPv2c、SNMPv3	SNMPv2c、SNMPv3
适用场景	小规模数据检索	大规模数据检索
配置难度	简单	较复杂
网络开销	高	低

常见陷阱04

- Max PDU size
- Max repetitions

```
# 每个请求的超时时间
timeout = "5s"
# SNMP Agent的版本
version = 2
# SNMP团体名
community = "public"
path = ["/opt/categraf/mibs/huawei/switch"]
translator = "gosmi"
agent_host_tag = "device_ip"
retries = 3
# GetBulk 每次可重复的最大重复数
max_repetitions = 20
```

#04

采集SNMP最佳实践

使用Categraf下SNMP插件

中国北京 2024.7.26

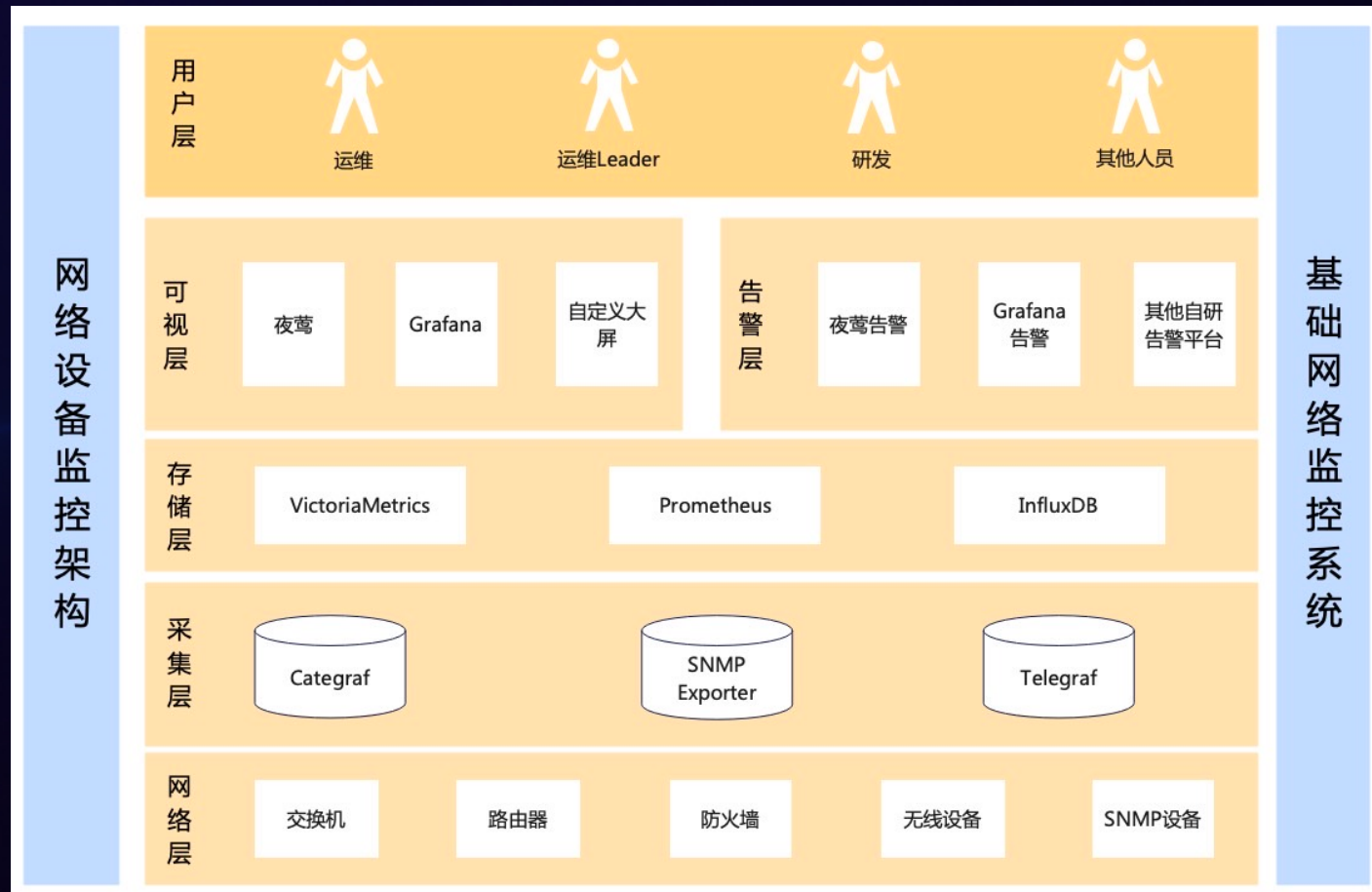


中國計算機學會
CHINA COMPUTER FEDERATION



网络架构

- 网络层开启 SNMP Agent
- 采集层可选择 Categraf、SNMP Exporter、Telegraf等
- 存储层可选择VictoriaMetrics、Prometheus、InfluxDB等
- 可视化可选择夜莺和Grafana，结合自身开发的前端大屏
- 告警层可选择夜莺告警、或者Grafana告警、或者其他自研的告警平台等



最佳实践

MIB库文件准备并上传到对应品牌管理目录中

品牌	设备类型	目录
华三	交换机/路由器/防火墙/无线	/opt/categraf/mibs/h3c/
华为	交换机/路由器/防火墙/无线	/opt/categraf/mibs/huawei/
思科	交换机/路由器/防火墙/无线	/opt/categraf/mibs/cisco/
公共MIB	公共依赖MIB	/opt/categraf/mibs/public/

最佳实践

根据类型、品牌、角色、架构层次的不同做采集配置文件的分类

品牌	网络Layer	角色	配置文件
华三	核心	交换机	snmp_h3c_switch_core.toml
华三	汇聚接入	无线	snmp_h3c_wireless_access.toml
华三	出口	防火墙	snmp_h3c_firewall_egress.toml
华为	核心	交换机	snmp_huawei_switch_core.toml
华为	汇聚接入	无线	snmp_huawei_wireless_access.toml
华为	出口	防火墙	snmp_h3c_firewall_egress.toml

最佳实践

- 配置文件采集前段配置
- 定义采集间隔和网络设备管理IP
- 自定义标签添加 labels
- SNMP 版本选择 v2c 还是 v3 版本
- MIB文件解析模式选择
- 网络设备管理IP标签名定义
- 重试次数和超时时间以及 GetBulk 最大重复次数设置

```
interval = 15

[[instances]]
agents = [
    "udp://172.16.100.10",
    "udp://172.16.100.11"
]

# metrics_name_prefix = "huawei_"

interval_times = 1
# 这里可以自定义指标标签
labels = { region = "beijing", role = "switch", brand = "huawei" }

timeout = "5s"
version = 2
community = "public"
path = ["/opt/categraf/mibs/huawei/switch"]
translator = "gosmi"
agent_host_tag = "device_ip"
retries = 3
max_repetitions = 15
```

最佳实践

- 配置文件实际网络指标采集配置
- 标量: `[[instances.field]]` 采集
- 表量: `[[instances.table]]` 采集整个表节点的 oid 字段, String 类型值无法作为指标存在
- 表量: `[[instances.table.field]]` 采集并加工表中的具体 oid 字段

```
[[instances]]
...

# 系统启动时间
[[instances.field]]
oid = ".1.3.6.1.2.1.1.3.0"
name = "uptime"

# 系统名称
[[instances.field]]
oid = ".1.3.6.1.2.1.1.5.0"
name = "deviceName"
is_tag = true

# 接口表信息
[[instances.table]]
name = "interface"
index_as_tag = true
inherit_tags = ["device_name"]

[[instances.table.field]]
oid = "IF-MIB::ifHighSpeed"
name = "ifHighSpeed"
is_tag = true

[[instances.table.field]]
oid = "IF-MIB::ifOperStatus"
name = "ifOperStatus"
```

最佳实践

```
# 保证 inputs.snmp 目录下只有当前调试测试配置文件
snmp_huawei_switch_core.toml

# 切换到root权限用户下操作测试 netsnmp + 环境变量
export MIBDIRS="/opt/categraf/mibs/huawei/switch:/opt/categraf/mibs/public"
# 执行categraf命令测试采集snmp 查看异常的输出
/opt/categraf/categraf --test --inputs snmp

# 切换到root权限用户下操作测试 gosmi + path 模式
# 直接执行 不需要设置临时环境变量 导入 MIB 文件读取目录
/opt/categraf/categraf --test --inputs snmp
```

最佳实践

```
root@alert:~# /opt/categraf/categraf --test --inputs snmp
2024/07/24 17:37:53 main.go:149: I! runner.binarydir: /opt/categraf
2024/07/24 17:37:53 main.go:150: I! runner.hostname: alert
2024/07/24 17:37:53 main.go:151: I! runner.fd_limits: (soft=1048576, hard=1048576)
2024/07/24 17:37:53 main.go:152: I! runner.vm_limits: (soft=unlimited, hard=unlimited)
2024/07/24 17:37:53 provider_manager.go:60: I! use input provider: [local]
2024/07/24 17:37:53 prometheus_agent.go:19: I! prometheus scraping disabled!
2024/07/24 17:37:53 ibex_agent.go:19: I! ibex agent disabled!
2024/07/24 17:37:53 agent.go:38: I! agent starting
2024/07/24 17:38:23 metrics_agent.go:317: I! input: local.snmp started
2024/07/24 17:38:23 agent.go:46: I! [*agent.MetricsAgent] started
2024/07/24 17:38:23 agent.go:49: I! agent started
17:38:43 snmp_icmp_up agent_hostname=alert brand=h3c device_ip=192.168.126.190
region=beijing role=wireless 1
17:38:43 snmp_icmp_rtt agent_hostname=alert brand=h3c device_ip=192.168.126.190
region=beijing role=wireless 41771.75
17:38:43 snmp_icmp_packet_loss agent_hostname=alert brand=h3c
device_ip=192.168.126.189 region=beijing role=wireless 0
17:38:43 snmp_uptime agent_hostname=alert brand=h3c device_ip=192.168.126.189
device_name=AC-01 region=beijing role=wireless 2217370962
```

最佳实践

```
root@alert:~# sudo vim /opt/categraf/conf/config.toml
```

```
...
```

```
[[writers]]
```

```
# VictoriaMetrics 时序库
```

```
url = "http://172.17.40.5:8429/api/v1/write"
```

```
...
```

```
[[writers]]
```

```
# Prometheus 时序库
```

```
url = "http://172.17.40.5:9090/api/v1/write"
```

```
...
```

感谢聆听

Thank you for listening