

金山办公基于 DeepFlow 的可观测性实践

肖圆 金山办公高级研发 私有化可观测平台负责人

CONTENTS

- 01 可观测建设现状
- 02 可观测落地与实践
- 03 可观测后续规划

可观测建设现状

- 架构调整：从k8s微服务架构逐渐演变为单元化架构
- 战略调整：可观测作为核心能力模块成为公司级项目

为配合架构演进需求，技术团队将基于 DeepFlow 可观测平台，在纯Docker环境（采用host网络模式）中构建新一代的全栈可观测体系，为架构转型提供坚实的技术保障。

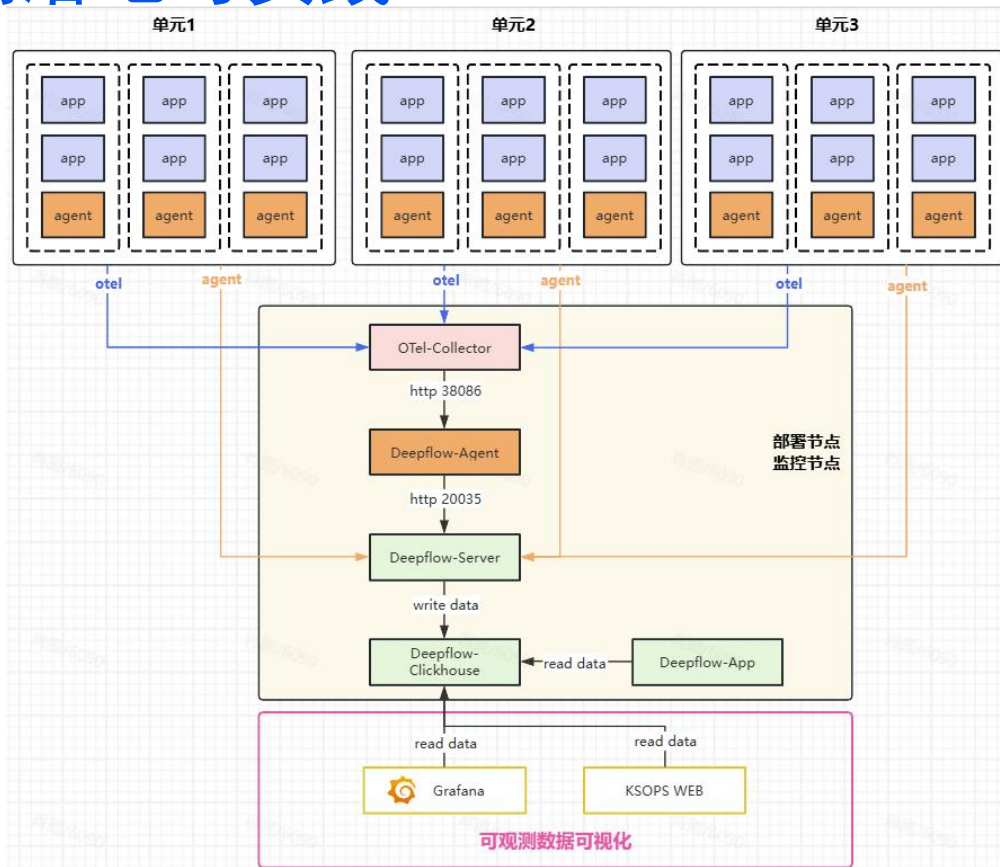
CONTENTS

- 01 可观测建设现状
- 02 可观测落地与实践
- 03 可观测后续规划

02 可观测落地与实践

部署架构(1)

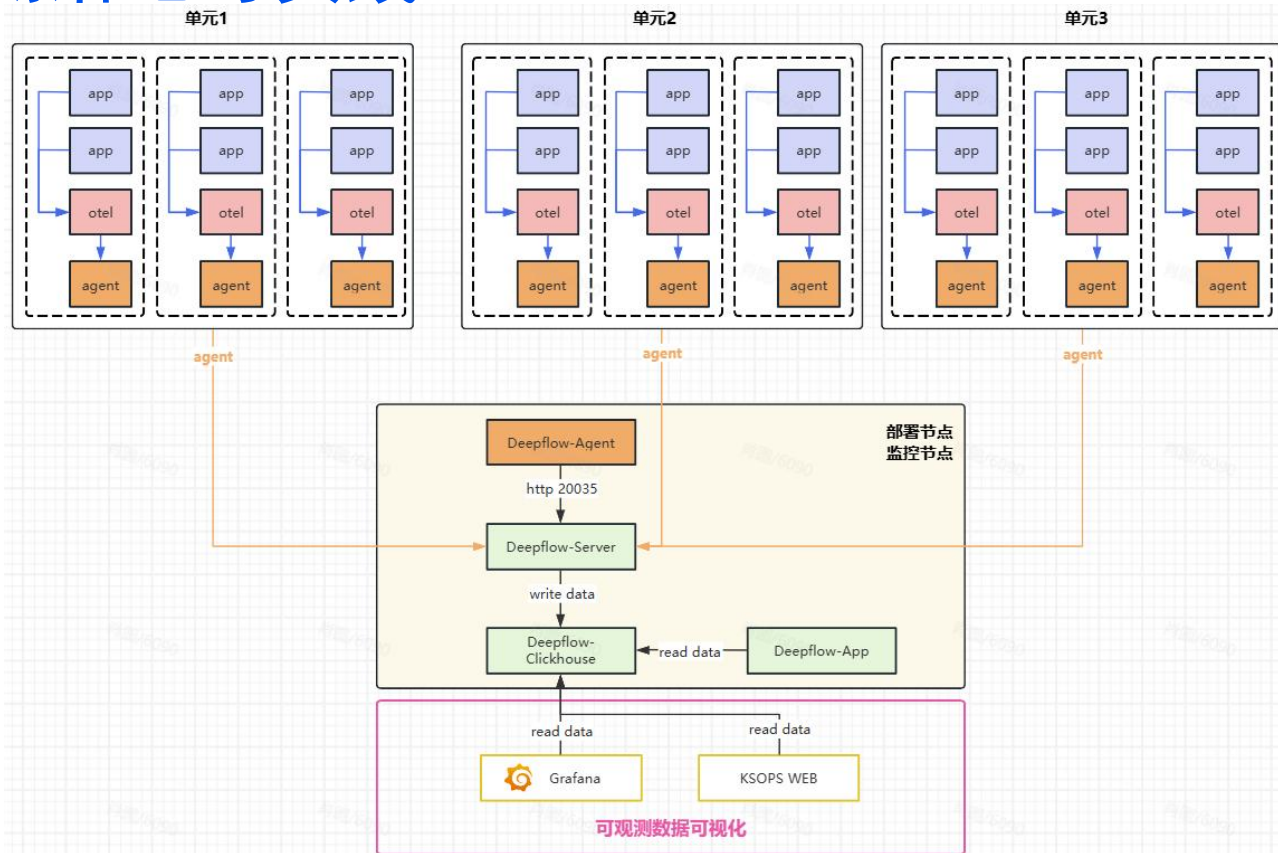
多otel sdk
|
单个collector
|
单个agent
|
单个server
|
单个ck



02 可观测落地与实践

部署架构(2)

单个otel sdk
|
单个collector
|
单个agent
|
单个server
|
单个ck



部署方式

server-docker-compose.yaml:

```
version: '3.8'
services:
  clickhouse:
    image: 'deepflow-clickhouse:v23.8.7.24'
    network_mode: host
    .....
  deploy:
    resources:
      limits:
        cpus: '4'
        memory: '32g'
  deepflow-server:
    image: 'reg.wps.lan:5000/wps/deepflow-server:v6.6'
    network_mode: host
    user: "65534"
    environment:
      DEEPFLOW_SERVER_RUNNING_MODE: STANDALONE
      K8S_NODE_IP_FOR_DEEPFLOW: 'x.x.x.x'
      K8S_POD_IP_FOR_DEEPFLOW: '::1'
      K8S_NAMESPACE_FOR_DEEPFLOW: deepflow
      K8S_NODE_NAME_FOR_DEEPFLOW: deepflow-host
      K8S_POD_NAME_FOR_DEEPFLOW: deepflow-server
    .....
```

```
deploy:
  resources:
    limits:
      cpus: '4'
      memory: '16g'
```

```
deepflow-app:
  image: 'deepflow-app:v6.6'
  network_mode: host
```

```
.....
deploy:
  resources:
    limits:
      cpus: '2'
      memory: '4g'
```

```
otel-collector:
  image: 'otel-collector:v0.101.0'
  network_mode: host
```

```
.....
deploy:
  resources:
    limits:
      cpus: '4'
      memory: '4g'
```

agent-docker-compose.yaml:

```
version: '3.2'
services:
  deepflow-agent:
    image: 'deepflow-agent:v6.6'
    privileged: true
    network_mode: host
    pid: host
    cap_add:
      - SYS_ADMIN
      - SYS_RESOURCE
      - SYS_PTRACE
      - NET_ADMIN
      - NET_RAW
      - IPC_LOCK
      - SYSLOG
    volumes:
      - "/data/kube/deepflow-agent/deepflow-agent.yaml:/etc/deepflow-agent/deepflow-agent.yaml:ro"
      - /sys/kernel/debug:/sys/kernel/debug:ro
      - /var/run/docker.sock:/var/run/docker.sock:ro
```

02 可观测落地与实践

部署配置

agent-group-config.yaml:

```
inputs:
  resources:
    workload_resource_sync_enabled: true
  proc:
    enabled: true
    process_matcher:
      - enabled_features:
          - ebpf.profile.on_cpu
        match_regex: ^(deepflow|docmini|mysqld|docs|node|nginx|docs-file-
cps|editserver|transcoder|cloudprovider|htmlserver|apiserver)
      only_in_container: false
  integration:
    enabled: true
    listen_port: 38086
  global:
    limits:
      max_memory: 2048
      max_millicpus: 2000
  circuit_breakers:
    relative_sys_load:
      trigger_threshold: 2.0
      recovery_threshold: 1.5
    metric: load15
```

```
communication:
  proxy_controller_port: 20035
  ingester_port: 20033
  proxy_controller_ip: "x.x.x.x"
  ingester_ip: "x.x.x.x"
processors:
  request_log:
    tag_extraction:
      tracing_tag:
        x_request_id:
          - X-Request-Id
    http_endpoint:
      match_rules:
        - keep_segments: 3
        url_prefix: ""
    obfuscate_protocols: [MySQL,Redis,Kafka]
  application_protocol_inference:
    enabled_protocols: [MySQL,Redis,Kafka,HTTP,HTTP2,gRPC]
```


02 可观测落地与实践

1、Docker 模式下无法采集容器资源信息

说明:

1. 当前 v6.6 版本中暂未加入对 Docker 环境的适配, 无法识别 Docker 容器的名称。
2. 目前对于 Docker 容器的识别可通过 eBPF 采集到的进程名称 (容器需使用宿主机网络) + IP + PORT

Request List												
Flame Graph	Status	Response Exception	Response Delay	Start time ↓	Client ip	Client Process	Server ip	Server Process	server_port	Data type	Enum(observation_p	
id-75193...	Success	--	26 μs	2025-06-24 10:36:27.997315	10.0.0.3	--	10.0.0.3	nginx	22307	eBPF	Server Process	
id-75193...	Client Error	Not Found	63 μs	2025-06-24 10:36:27.997207	10.0.0.3	--	10.0.0.3	calico-w	22104	eBPF	Server Process	
id-75193...	Success	--	84 μs	2025-06-24 10:36:27.997204	10.0.0.3	--	10.0.0.3	nginx	22155	eBPF	Server Process	
id-75193...	Client Error	Not Found	395 μs	2025-06-24 10:36:27.997019	10.0.0.3	keepalived	10.0.0.3	--	22104	eBPF	Client Process	
id-75193...	Client Error	Not Found	87 μs	2025-06-24 10:36:27.996950	10.0.0.3	--	10.0.0.3	editproxy	20024	eBPF	Server Process	
id-75193...	Success	--	486 μs	2025-06-24 10:36:27.996937	10.0.0.3	keepalived	10.0.0.3	宿主 IP	22104	eBPF	容器进程	容器端口
id-75193...	Client Error	Not Found	174 μs	2025-06-24 10:36:27.996905	10.0.0.3	keepalived	10.0.0.3	--	20024	eBPF	Client Process	
id-75193...	Success	--	627 μs	2025-06-24 10:36:27.996855	10.0.0.3	keepalived	10.0.0.3	--	22307	eBPF	Client Process	
id-75193...	Success	--	50 μs	2025-06-24 10:36:27.996850	10.0.0.3	--	10.0.0.3	ksysd	22270	eBPF	Server Process	
id-75193...	Success	--	234 μs	2025-06-24 10:36:27.996809	10.0.0.3	keepalived	10.0.0.3	--	22270	eBPF	Client Process	

02 可观测落地与实践

2、IPV6 环境部署适配性问题

说明：部署后发现 Server 端没有 controller ip，经排查后发现 Server 配置文件中存在多处配置格式不统一，不太易用。与社区及时沟通后将配置格式统一为后解决问题。

```
also, kubernetes_watch_policy: 0, group_id: g-c46134c14b) not found in cache. NAME:deepflow-agent-ce
750146118
2025-06-20 17:21:24.208 [INFO] [trisolaris.vtap] vtap.go:1368 start vtap register
2025-06-20 17:21:24.208 [ERRO] [trisolaris.vtap] vtap.go:223 config NodeIP is null
2025-06-20 17:21:24.208 [ERRO] [trisolaris.vtap] vtap.go:1343 controller not found region
2025-06-20 17:21:24.208 [INFO] [trisolaris.vtap] vtap.go:1375 end vtap register
2025-06-20 17:21:24.234 [INFO] [tagrecorder] dictionary.go:81 tagrecorder update ch dictionary
```

```
controller:
  election-name: deepflow-server
  election-namespace: deepflow
  grpc-max-message-length: 104857600
  grpc-port: 20035
  kubeconfig:
  listen-port: 20417
  clickhouse:
    database: flow_tag
    host: "[2408:8656:22df:ff01::14:fe06]"
    port: 19000
    user-name: default
    user-password:
  mysql:
    auto_increment_increment: 1
    database: deepflow
    host: "2408:8656:22df:ff01::14:fe06"
    port: 4306
    timeout: 30
    user-name: root
    user-password:
```

改之前配置

```
controller:
  election-name: deepflow-server
  election-namespace: deepflow
  grpc-max-message-length: 104857600
  grpc-port: 20035
  kubeconfig:
  listen-port: 20417
  clickhouse:
    database: flow_tag
    host: "2408:8656:22df:ff01::14:fe06"
    port: 19000
    user-name: default
    user-password:
  mysql:
    auto_increment_increment: 1
    database: deepflow
    host: "2408:8656:22df:ff01::14:fe06"
    port: 4306
    timeout: 30
    user-name: root
    user-password:
```

改之后配置

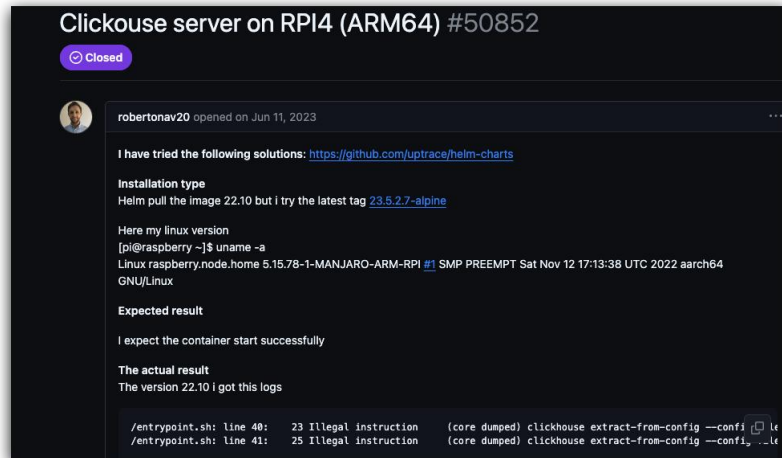
02 可观测落地与实践

3、ARM 环境部署时，ClickHouse 需要单独拉 ARM 镜像

说明：DeepFlow 自身相关的组件镜像均支持 ARM 环境，其他中间件 MySQL 与 CK 需要从对应社区同步镜像。

DeepFlow 默认提供的是 X86 镜像，解决方式分为两种：

1. 去对应社区拉取镜像：例如可去鲲鹏官网查询同步的 ARM 镜像版本
2. 使用自建 CK 环境或自定义构建 ARM 版本镜像。自定义镜像可参考以下方式：
 - <https://github.com/ClickHouse/ClickHouse/issues/50852>
 - https://www.hikunpeng.com/document/detail/zh/kunpengcdfs/ecosystemEnable/Docker/kunpengdocker_03_0011.html



02 可观测落地与实践

4、业务端口与 Server 端口冲突

说明：部分 Server 端口与业务端口冲突，导致业务无法部署。可在 server 配置中可自定义各个模块所用端口：

- <https://github.com/deepflowio/deepflow/blob/main/server/server.yaml>

```
controller:
  ## controller http listenport
  #listen-port: 20417
  #listen-node-port: 30417
  ## grpc server port
  #grpc-port: 20035
  #ssl-grpc-port: 20135
  #agent_ssl_cert_file: /etc/ssl/server.key
  #agent_ssl_key_file: /etc/ssl/server.pem
  #ingester-port: 20033
  #grpc-node-port: 30035

  ## controller listening port
  #controller-port: 20035

  ## stats collect interval(unit: s)
  # stats-interval: 10

  ## The listening port used by Ingester to receive data
  #listen-port: 20033
  #
  ## The listening port used by Ingester to handle datasource API
  #datasource-listen-port: 20106
```

02 可观测落地与实践

5、大版本回退闭坑

说明：大版本升级的注意事项

- 1、社区不同大版本 server 和 agent 不可兼容
- 2、大版本升级前一定要做测试，因为升级后如果有问题不能支持回退



彭斌

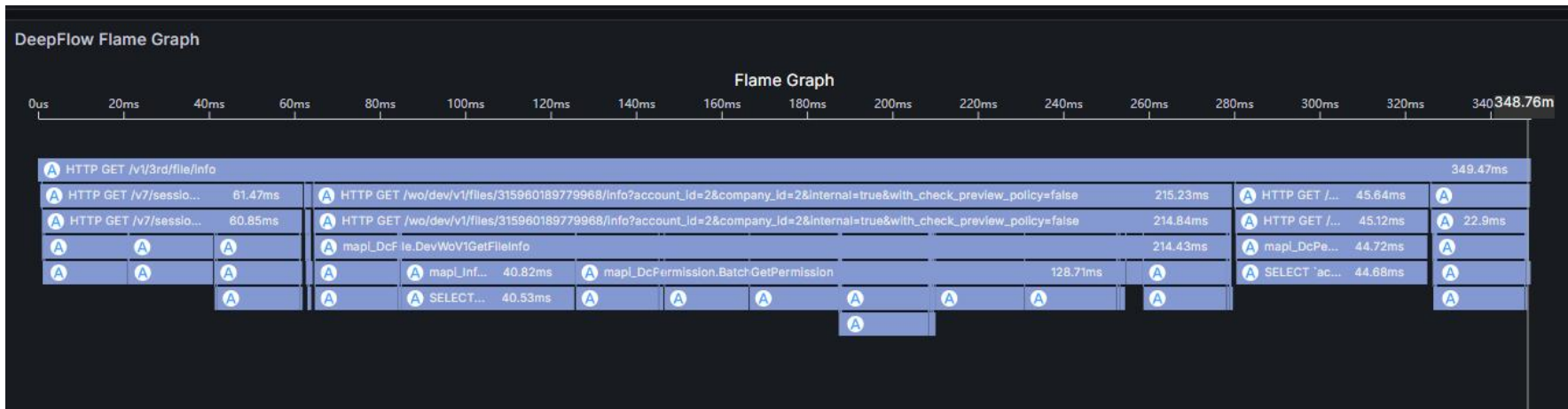
主要是大版本升级，如果有现成的测试环境，还是先测试一下比较好，

正常来说我们只是不支持版本回退，但是升级的话倒也没啥问题

肖圆：好，看来有坑？🤔

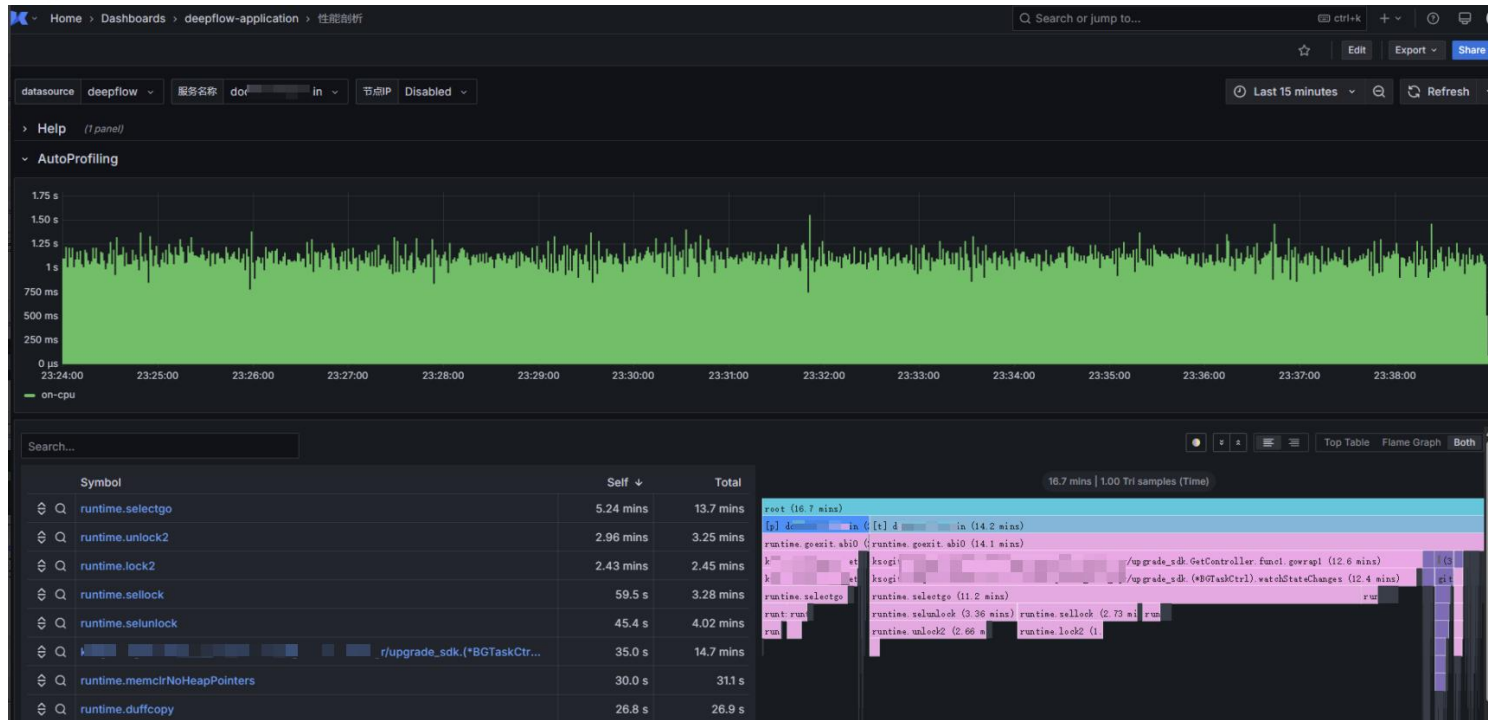
02 可观测落地与实践

阶段性成果(一): 函数级链路数据



02 可观测落地与实践

阶段性成果(二): 案例-微服务占用cpu不释放



02 可观测落地与实践

阶段性成果(三): 案例-MySQL访问延迟

Home > Dashboards > deepflow-application > 服务列表

Data Source: deepflow, auto_service, Any, signal_source: eBPF

Service List		Request Rate	Delay Avg	Delay P75	Delay P90	Delay P99	Client Error Ratio	Server Error Ratio	Signal Source
Auto Service			554.94 ms	1.00 s	1.01 s	1.01 s	1.62%	97.28%	eBPF
169.254.30.52		0.83 req/s		1.00 s	1.00 s	1.00 s	0.00%	41.27%	eBPF
169.254.20.78		0.37 req/s	336.71 ms			1.38 ms	0.00%	30.11%	eBPF
169.254.20.17		0.78 req/s	435.54 μs	600.97 μs	713.58 μs	1.00 s	45.18%	6.69%	eBPF
127.0.0.1		1.02 req/s	190.76 ms	1.00 s			49.25%	0.59%	eBPF
169.254.20.99		94.66 req/s	113.41 ms	1.47 ms	14.08 ms	1.01 s	0.00%	0.10%	eBPF
10.13.1.73		6.68 req/s	658.60 μs	669.18 μs	670.94 μs	697.70 μs	0.00%	0.08%	eBPF
10.13.1.72		1.08K req/s	367.62 ms	500.61 ms	500.86 ms	1.01 s	3.11%	0.00%	eBPF
10.13.6.17		1.53K req/s	112.25 ms	500.37 ms	500.44 ms	500.79 ms	0.86%	0.00%	eBPF
169.254.20.31		3.52K req/s	163.46 ms	500.74 ms	500.84 ms	501.14 ms	4.48%	0.00%	eBPF
		0.16 req/s	1.97 ms	2.36 ms	4.08 ms	8.00 ms	0.00%	0.00%	eBPF

Endpoint List

Endpoint	Signal Source	Delay Avg	Delay P75	Delay P90	Delay P99	Client Error Ratio	Request Sum	Server Error Ratio	ip_1	server_icon_id	server_node_type	server_re
/office/v5/dev	eBPF	106.20 ms	17.27 ms	147.64 ms	1.64 s	0.00%	59.00	0.00%	169.254.20.40	0	ip	169.254.2
wps_mail_puller_v2-3	eBPF	500.37 ms	500.72 ms	500.99 ms	501.45 ms	0.00%	632.00	0.00%	10.13.1.73	0	ip	10.13.1.73

Home > Dashboards > deepflow-application > 链路追踪

Data Source: deepflow, ip, Any, I7.protocol: HTTP + Any, data_type: eBPF + OTel, status: Any, service: Disabled, request_id: *, trace_id: *

request_resource: *, endpoint: *, deepflow_tracing_id: Enter variable value

Host IP	eBPF Request ID	Otel Request ID	Trace ID	Service Name	Protocol	Request type	Request domain	Request resource	Endpoint	Status	Request c	Response Exception	Response Delay
10.13.1.73	--	1416040bbb74f9469: 176fcd57abf40e55c1f	doc	5-iam	HTTP	GET	dc.....etaw	/login/api/v1/config?comp	V1GetLoginCfg	Server Error	2	dial tcp 169.254.20.12:3306: i/o timeout	1.00 s
10.13.1.73	--	1416040bbb74f9499: 3dd9034e3397ecf27	do	5-iam	HTTP	GET	dc.....w	/login/api/v1/config?comp	V1GetLoginCfg	Server Error	2	dial tcp 169.254.20.12:3306: i/o timeout	1 s
--	--	--	--	--	HTTP	POST	localhost:8428	/api/v1/query?query=cont	/api/v1/query	Success	200	--	30.1 ms
--	--	--	--	--	HTTP	POST	localhost:8428	/api/v1/query?query=cont	/api/v1/query	Success	200	--	30.1 ms

02 可观测落地与实践

阶段性成果(四): 案例-Redis访问延迟

Home > Dashboards > deepflow-application > 链路追踪

Data Source: deepflow | ip: Any | I7_protocol: Any | data_type: OTel | status: Any | service: Disabled | request_id: * | trace_id: *

request_resource: * | endpoint: * | deepflow_tracing_id: Enter variable value

Host IP	eBPF Request ID	Otel Request ID	Trace ID	Service Name	Protocol	Request type	Request domain	Request resource	Endpoint	Status	Request c	Response Exception	Response Delay
10.13.1.73	--	0001070bbc65e8240	4ebad2562ae45c15b	doc	Redis	--	--	get;	redis	Server Error	2	i/o timeout;	8.00 s
10.13.1.73	--	0001070bbc65e8240	b97e47ecd36a13aa31	doc	Redis	--	--	get;	redis	Server Error	2	i/o timeout;	8.00 s
10.13.6.17	--	**	0018cc214f5b12b1158	doc	Redis	--	--	ping;	redis	Server Error	2	i/o timeout;	8.00 s
10.13.6.17	--	**	fed330afbe71b009ee	doc	Redis	--	--	cluster;	redis	Server Error	2	i/o timeout;	9.00 s
10.13.1.73	--	0001060bbc65e8210	32df5db852ec98eca	doc	Redis	--	--	get;	redis	Server Error	2	dial tcp 10.13.1.72:853	10.0 s
10.13.6.17	--	184A31A75881CFF8	441e4d7eb645914ebf	doc	N/A	--	--	SELECT 'company_id','cre db		Success	1	--	1.18 ms
10.13.1.73	--	184A803BE9DD119E	92a0884e53004a4c5	doc	N/A	--	--	SELECT 'company_id','cre db		Success	1	--	1.90 ms

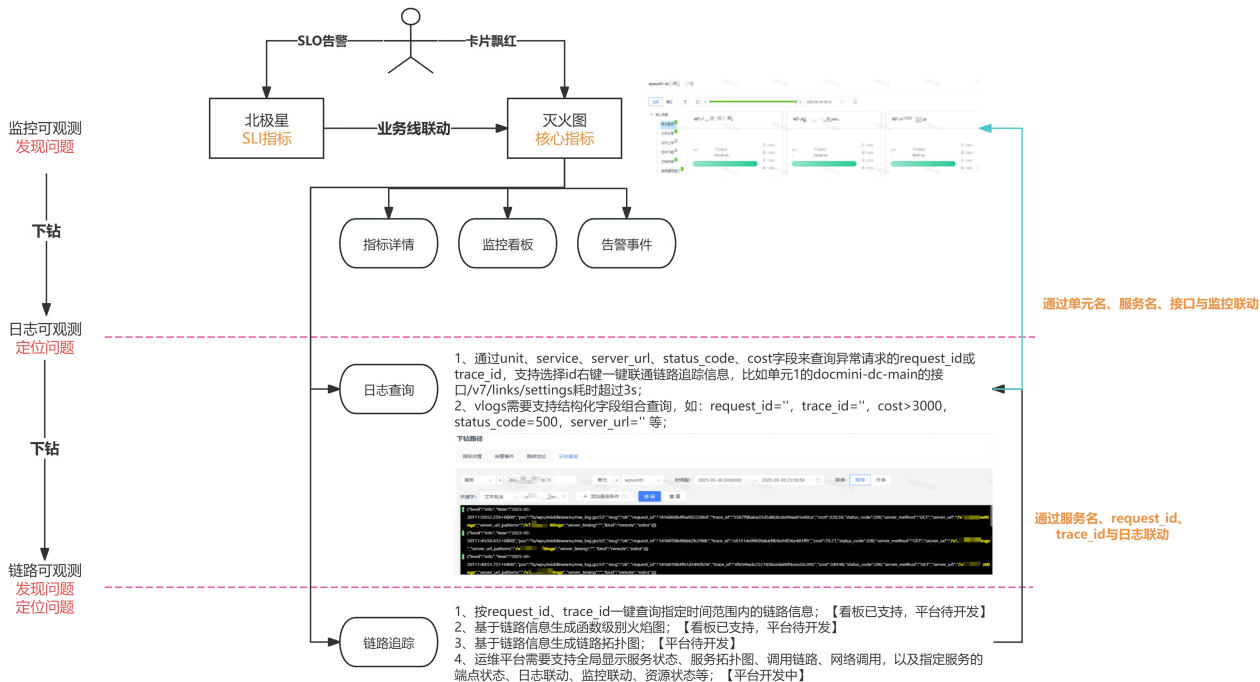
02 可观测落地与实践

阶段性成果(五): 系统适配

- 操作系统: centos、redhat、ubuntu、suse、uos、kylin
- cpu架构: amd64、arm64
- IP栈: ipv4、ipv6

02 可观测落地与实践

阶段性成果(六): 数据联动



CONTENTS

- 01 可观测建设现状
- 02 可观测落地与实践
- 03 可观测后续规划

可观测后续规划

1、应用侧全链路接入

- 推动各业务线完成OpenTelemetry标准化接入改造
- 建立接入进度跟踪机制，确保Q3 100%覆盖核心业务系统

2、统一观测平台升级

- 开发新一代可观测分析平台，实现四维数据联动：
 - Metrics指标分析
 - Logs日志检索
 - Traces调用链追踪
 - Profiling性能剖析
- 构建多维度数据关联分析能力

3、有效性验证体系

- 设计混沌工程实验方案
- 通过故障注入验证观测系统的：
 - 指标覆盖完整性
 - 告警准确率
 - 根因定位效率

4、引入AIOps能力

- 部署智能分析Agent实现：
 - 7×24小时自动化巡检
 - 异常模式智能识别
 - 故障根因辅助分析
- 建立观测数据知识图谱



THANKS
感谢您的观看
Thank you for
watching

肖圆 金山办公