1、coolbpf server镜像说明

coolbpf镜像可以提供以下服务:

- surftrace数据在线解析服务
- lcc bpf.c文件远程编译服务
- btf文件在线获取服务

要运行coolbpf镜像，需要具备以下条件：

1. 目标实例支持docker
2. 目标实例预留100G左右的磁盘空间（存放btf/db文件）
3. 如果要实时更新btf/db，需要支持访问pylcc.openanolis.cn
4. surftrace >=0.6.3 pylcc >= 0.2.2

2、搭建coolbpf 编译服务

我们以在192.168.22.4 实例上搭建lbccompile服务上搭建服务为例。

2.1、同步db/btf：

在实例上创建目录，如/root/1ext/hive，并在该目录下，同步db/btf数据源：

```
rsync -av pylcc.openanolis.cn::pylcc/btf .
rsync -av pylcc.openanolis.cn::pylcc/db .
rsync -av pylcc.openanolis.cn::pylcc/header .
```

可以将rsync 放到crontab 定时任务中去，与远端定期保持同步。

2.2、启动容器

```
docker run  --entrypoint="/bin/bash" --name surfd  -v /root/1ext/hive:
/home/hive -p 7655:7655 -itd registry.cn-hangzhou.aliyuncs.com/alinux/
coolbpf /home/lbc/run.sh 127.0.0.1
```

2.3、验证

```
export LBC_SERVER=192.168.22.4
surftrace 'p _do_fork'

echo 'p:f0 _do_fork' >> /sys/kernel/debug/tracing/kprobe_events
echo 1 > /sys/kernel/debug/tracing/instances/surftrace/events/kprobes/
f0/enable
echo 0 > /sys/kernel/debug/tracing/instances/surftrace/options/stacktr
ace
echo 1 > /sys/kernel/debug/tracing/instances/surftrace/tracing_on
 <...>-39643 [002] .... 2073472.895508: f0: (_do_fork+0x0/0x3a0)
 staragentd-27114 [003] .... 2073472.977518: f0: (_do_fork+0x0/0x3a0)
 <...>-39662 [001] .... 2073472.980098: f0: (_do_fork+0x0/0x3a0)
 ……
```

hello.py

```
import time
from pylcc.lbcBase import ClbcBase

bpfPog = r"""
#include "lbc.h"

SEC("kprobe/wake_up_new_task")
int j_wake_up_new_task(struct pt_regs *ctx)
{
    struct task_struct* parent = (struct task_struct *)PT_REGS_PARM1(ctx
    bpf_printk("hello lcc, parent: %d\n", _(parent->tgid));
    return 0;
}

char _license[] SEC("license") = "GPL";
"""

class Chello(ClbcBase):
    def __init__(self):
        super(Chello, self).__init__("hello", bpf_str=bpfPog)
        while True:
            time.sleep(1)


if __name__ == "__main__":
    hello = Chello()
    pass
```

执行：```

python hello.py

remote server compile success. ^CTraceback (most recent call last): File "hello.py", line 26, in hello = Chello() File "hello.py", line 22, in **init** time.sleep(1) KeyboardInterrupt

[root@localhost.localdomain /root/1ext/hive]

cat /sys/kernel/debug/tracing/trace_pipe

```
     python3-1770  [000] .... 2073737.147865: 0: hello lcc, parent: 4009
       <...>-40096 [003] .... 2073737.163631: 0: hello lcc, parent: 4009
       <...>-40096 [003] .... 2073737.166262: 0: hello lcc, parent: 4009
       <...>-40096 [003] .... 2073737.166328: 0: hello lcc, parent: 4009
       <...>-40096 [003] .... 2073737.166376: 0: hello lcc, parent: 4009
       <...>-40096 [003] .... 2073737.166437: 0: hello lcc, parent: 4009
```