A-Tune sig meeting

致力于调优领域相关技术探索与AI辅助的性能分析

openEuler

议程

1. A-Tune介绍及未来演进
2. 近一年社区的发展
3. kafka集群调优实例分享
4. 服务化网站分享
5. A-Tune-BPF-Collection介绍
6. 未来计划

openEuler

近一年社区的发展


2021 Milestone
sig组
新增应用
https://gitee.com/openeuler/community/
https://gitee.com/openeuler/A-Tune-BPF-Collection
沟通交流
与高校师生合作交流
https://lab.huaweicloud.com/testdetail_501
A-Tune应用
发布新版本
https://gitee.com/openeuler/A-Tune/releases
性能调优
mysql/nginx...
使用A-Tune工具对多款应用进行调优
内部创新
A-Tune服务化
对接PerfStudio网页，实现页面控制调优
外部联创
中国移动云能力中心创新团队
mysql基于BCLinux操作系统的性能调优

服务化网站分享

- https://perfstudio.server.huawei.com/

- UI v2.0 敬请期待……
  - https://gitee.com/openeuler/A-Tune-UI
  - http://localhost:8080/#/
  - 计划发布时间：930 （openEuler-22.09创新版本）

未来计划

A-Tune项目

- 解决现有社区issue
- 增加AI调优算法
- 加强集群调优功能
- 适配更多的使用场景
- 持续探索在线动态调优
- ……

性能调优

持续进行性能调优，计划对openeuler操作系统上运行的视频转码应用(e.g: ffmpeg)、web场景(e.g: tomcat)、大数据场景等应用进行调优。

社区建设

- 版本发布计划
- 与外部客户合作联创
- ……

沟通交流

暑期2022、sig meeting、openEuler summit、邮件列表……

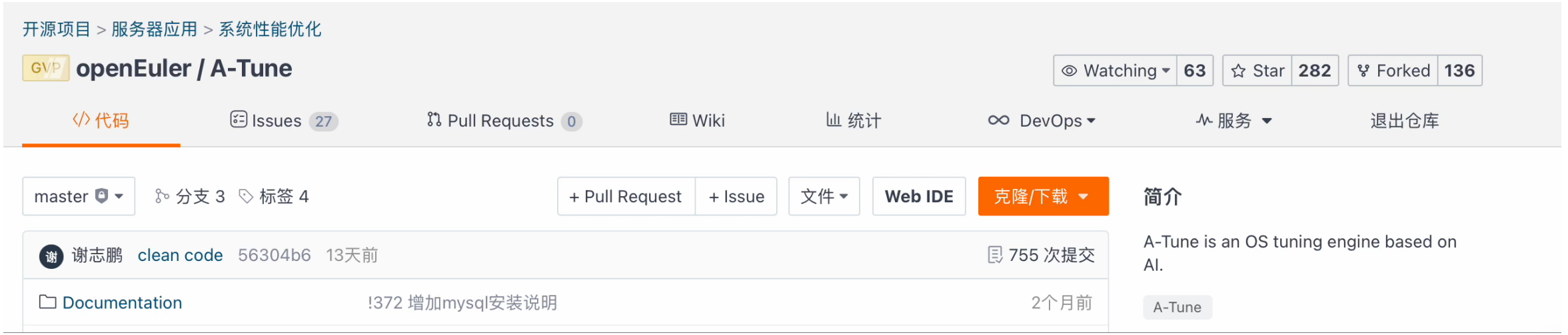
期待您的加入

A-Tune：https://gitee.com/openeuler/A-Tune

A-Tune-BPF-Collection：https://gitee.com/openeuler/A-Tune-BPF-Collection

A-Tune-Collector：https://gitee.com/openeuler/A-Tune-Collector

A-Tune-UI：https://gitee.com/openeuler/A-Tune-UI



openEuler

openEuler

Kafka集群调优案例分享

胡彬

openEuler

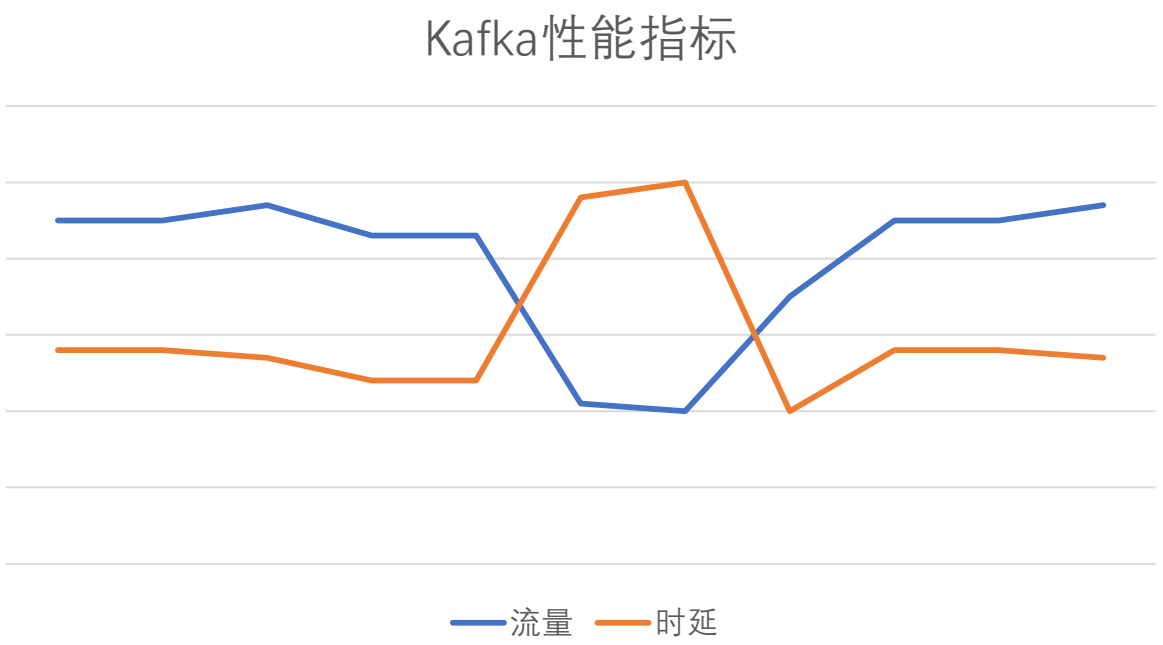
目录

openEuler

背景

kakfa集群存在性能突降现象

- Buffer 写操作被 fsync IO 阻塞严重导致时延增加
- 磁盘 IO await 耗时增加
- 刷脏页策略存在优化空间



确定调优目标

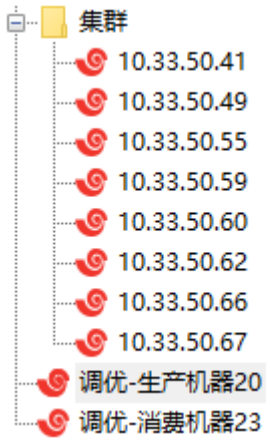
性能指标

- 生产者tps均值
- 时延均值
- 消费者tps均值
- 消费者tps最低值

```
2022-04-01 09:40:28.651:INFO:oejs.AbstractConnector:main: Started
2022-04-01 09:40:28.651:INFO:oejs.Server:main: Started @2344ms
TAG: producer tips = 168990 send Time is 286 ms
TAG: producer tips = 109047 send Time is 279 ms
TAG: producer tips = 108098 send Time is 283 ms
TAG: producer tips = 106259 send Time is 287 ms
TAG: producer tips = 107068 send Time is 284 ms
TAG: producer tips = 110351 send Time is 276 ms
TAG: producer tips = 108889 send Time is 279 ms
TAG: producer tips = 218817 send Time is 141 ms
```

测试方式

- 1生产者+8节点集群+1消费者
- 消息体1K大小左右，末尾添加一定的随机字符串，模拟业务实际场景
- 10分钟持续消息分发



openEuler

确定调优目标

待调优参数

集群节点：刷脏页策略相关参数

- 内核参数 vm.dirty_background_ratio
  - 可取值范围：0–100 整数
- 内核参数 vm.dirty_background_bytes
  - 可取值范围：0–系统可用内存字节数

使用A-Tune进行调优

编写调优脚本

- Benchmark 脚本：
  - 运行测试，并输出指标
- Tuning Client yaml 脚本
  - 提供从benchmark输出提取指标的方法
  - 定义迭代轮数及算法
- Tuning Server yaml 脚本
  - 定义待调节项，包括设置、获取方法

使用A-Tune进行调优

编写调优脚本

Benchmark 脚本

- 启动生产者及消费者进程
- 10min之后 kill 生产者及消费者进程
- 根据生产者和消费者日志计算性能指标并输出

```
# run test
cd /opt/huawei/1.3.13.100/KOPsdv2dmq-demo-1.3.13.100/bin; sh start.sh
ssh 10.33.50.23 'source /etc/profile; cd /opt/huawei/1.3.13.100/liaofkafkaconsumer/bin; sh start.sh'

# wait for a period of time
totaltime=10
for i in $(seq 1 $totaltime); do
echo "test running progress: ($i / $totaltime) minutes..."
sleep 60
done

# kill test programs
echo 'kill producer...'
pid=`ps aux | grep dmq-demo | grep -v grep | awk '{print $2}' | head -1`
[ "$pid" == "" ] && echo 'producer is not running!'
[ "$pid" != "" ] && echo "producer pid is $pid" && kill -9 $pid
echo 'kill producer done'
ssh 10.33.50.23 'sh /home/A-Tune/kill.sh' 2>/dev/null

# extract test result
echo "producer_tps: `cat /opt/huawei/1.3.13.100/KOPsdv2dmq-demo-1.3.13.100/logs/nohup.out | grep 'TAG: producer' |
echo "delay: `cat /opt/huawei/1.3.13.100/KOPsdv2dmq-demo-1.3.13.100/logs/nohup.out | grep 'TAG: producer' | tail -
ssh 10.33.50.23 'sh /home/A-Tune/get-result.sh' 2>/dev/null
```

使用A-Tune进行调优

编写调优脚本

Tuning Client yaml 脚本

- 指定benchmark脚本
- 从benchmark输出中提取相关性能指标
- 设置各个指标的调优权重
- 迭代轮数及调优算法

```
benchmark : "sh /root/A-Tune-master/examples/tuning/kafka/kafka_bench.sh"
evaluations :
  -
    name: "producer_tps"
    info:
        get: "echo '$out' | grep 'producer_tps:' | awk '{print $2}'"
        type: "negative"
        weight: 25
  -
    name: "delay"
    info:
        get: "echo '$out' | grep 'delay:' | awk '{print $2}'"
        type: "positive"
        weight: 25
  -
    name: "comsumer_tps"
    info:
        get: "echo '$out' | grep 'consumer_tps:' | awk '{print $2}'"
        type: "negative"
        weight: 25
  -
    name: "consumer_min_tps"
    info:
        get: "echo '$out' | grep 'consumer_min_tps:' | awk '{print $2}'"
        type: "negative"
        weight: 25
```

使用A-Tune进行调优

编写调优脚本

Tuning Server yaml 脚本

- 待调优参数 Get/Set 方法（通过 ssh 远程控制集群节点）
- 调优参数类型
- 调优参数范围

```
value=$1
ipaddrs=(10.33.50.41 10.33.50.49 10.33.50.55 10.33.50.59 10.33.5
for ipaddr in "${ipaddrs[@]}";
        do echo "set value for $ipaddr";
        ssh $ipaddr "sysctl -w vm.dirty_background_ratio=$value"
done
```

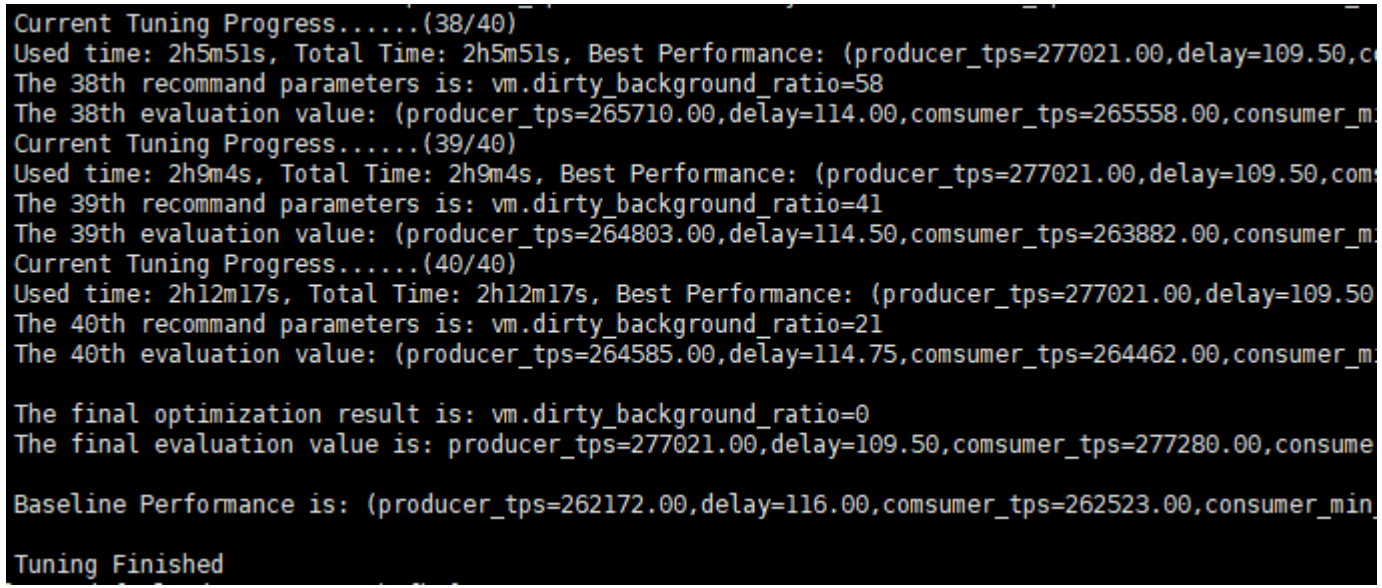
```
object :
  -
    name : "vm.dirty_background_bytes"
    info :
        desc : "Contains the amount of dirty memory at which
        get : "sh /home/A-Tune/get-bytes.sh"
        set : "sh /home/A-Tune/set-bytes.sh $value"
        needrestart : "false"
        type : "discrete"
        scope :
          - 0
          - 107374182400
        step : 524288000
        items :
        dtype : "int"
```

```
object :
  -
    name : "vm.dirty_background_ratio"
    info :
        desc : "Contains the amount of dirty memory at which
        get : "sh /home/A-Tune/get-ratio.sh"
        set : "sh /home/A-Tune/set-ratio.sh $value"
        needrestart : "false"
        type : "discrete"
        scope :
          - 0
          - 100
        step : 1
        items :
        dtype : "int"
```

使用A-Tune进行调优

开始Tuning调优

- sh prepare.sh
- atune-adm tuning --project kafka2 --detail kafka_client.yaml



调优结果

最优参数

设置 vm.dirty_background_ratio=0 为最优配置。

相比原始默认值 vm.dirty_background_ratio=10 提升约5%。

测试编号	vm.dirty_background_ratio	producer_tps	delay	comsumer_tps	consumer_min_tps	与基线对比（vm.dirty_background_ratio=10）
1	42	265260	114.5	265469	263110	1.30%
2	72	264244	115	264263	262576	0.93%
3	0	274160	110.5	274483	272992	4.88%
4	30	264252	115	263731	260751	0.70%
5	14	263078	115.5	263176	258518	0.21%
			…			
37	83	262195	115.75	262334	260910	0.23%
38	58	265710	114	265558	263740	1.53%
39	41	264803	114.5	263882	260481	0.85%
40	21	264585	114.75	264462	263240	1.10%
最优参数	0	277021	109.5	277280	275089	5.86%

测试编号	vm.dirty_background_bytes	vm.dirty_background_bytes（转换为GB）	producer_tps	delay	comsumer_tps	consumer_min_tps	与基线对比（vm.dirty_background_bytes=0 ）
1	71827456000	68.5	264849	114.75	264544	262856	-5.10%
2	98041856000	93.5	265134	114.5	265108	258817	-5.37%
3	524288000	0.5	267192	113.25	266524	264095	-4.21%
4	46137344000	44.0	263734	115	264337	262851	-5.29%
5	20971520000	20.0	266938	113.5	266748	266021	-4.08%
				…			
37	20971520000	20.0	267196	113.5	266393	262719	-4.42%
38	5242880000	5.0	265734	114	265755	264213	-4.59%
39	64487424000	61.5	267550	113.5	267423	266640	-3.90%
40	68157440000	65.0	265720	114	265405	263459	-4.70%
最优参数	64487424000	61.5	267550	113.5	267423	266640	-3.90%

openEuler

A-Tune BPF Collection

A-Tune + BPF program tuning kernel

openEuler

Background

当前计算机系统中OS开销比较高，内核自身执行有大量的开销，称之为“OS tax”。

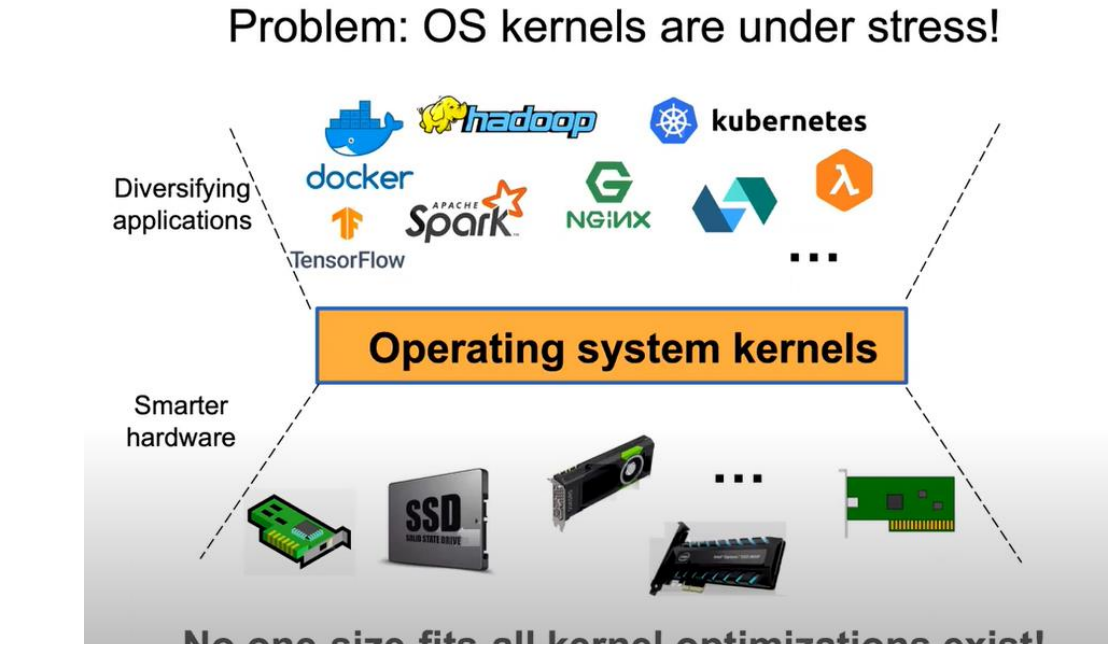
有两方面原因：

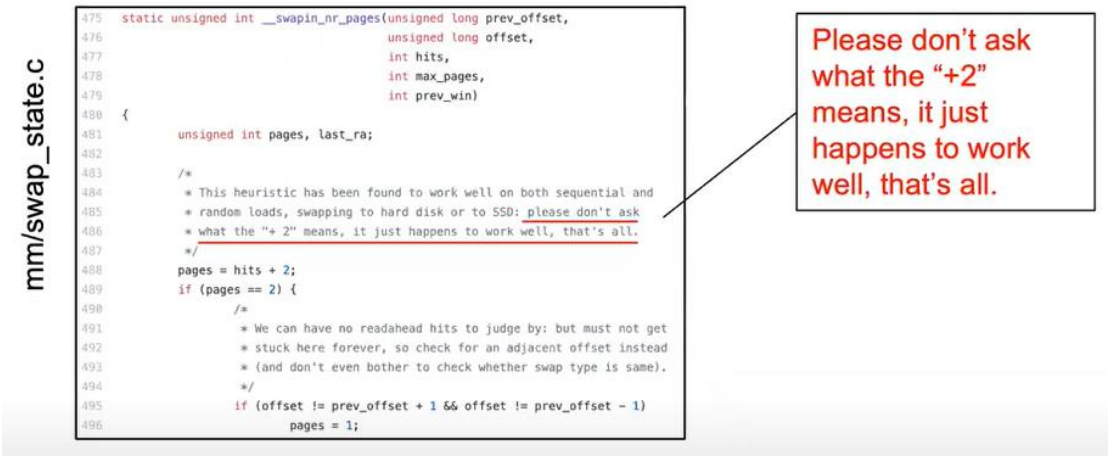
1、启发式硬编码设计，基于专家经验，而非数据驱动

2、面向通用场景设计，需要通过调优才能达到最优

发展趋势：

- 操作系统需要支持各种不同应用和硬件，没有一个通用的统一优化策略能对所有场景都适用
- 硬件发展速度要比软件更快，会导致系统中原有的优化策略在新硬件上表现达不到预期效果





Case Study – Spark file read

Spark读操作的数据类型：

- 元数据blk_*.meta：2KB ~ 3MB，读取粒度：8KB
- 块数据blk_*: 2KB ~ 256MB，读取粒度：1MB

Linux readahead:

- Pro: 文件顺序读场景下，会预先将后续数据预读入pagecache，后续读操作直接从pagecache中读取数据，从而减少磁盘I/O，提升文件读性能。
- Con: 随后的读操作不是预读的内容，让pagecache中的数据无用，仍需要进行磁盘I/O，导致“文件预读放大”

Spark读操作特征分析：

- 元数据的顺序读约为10%，但是读取粒度小，一次预读更多的元数据，能够减少元数据I/O
- 块数据的顺序读约为10%，大量随机读中夹杂少量连续的顺序读操作，且读取粒度大

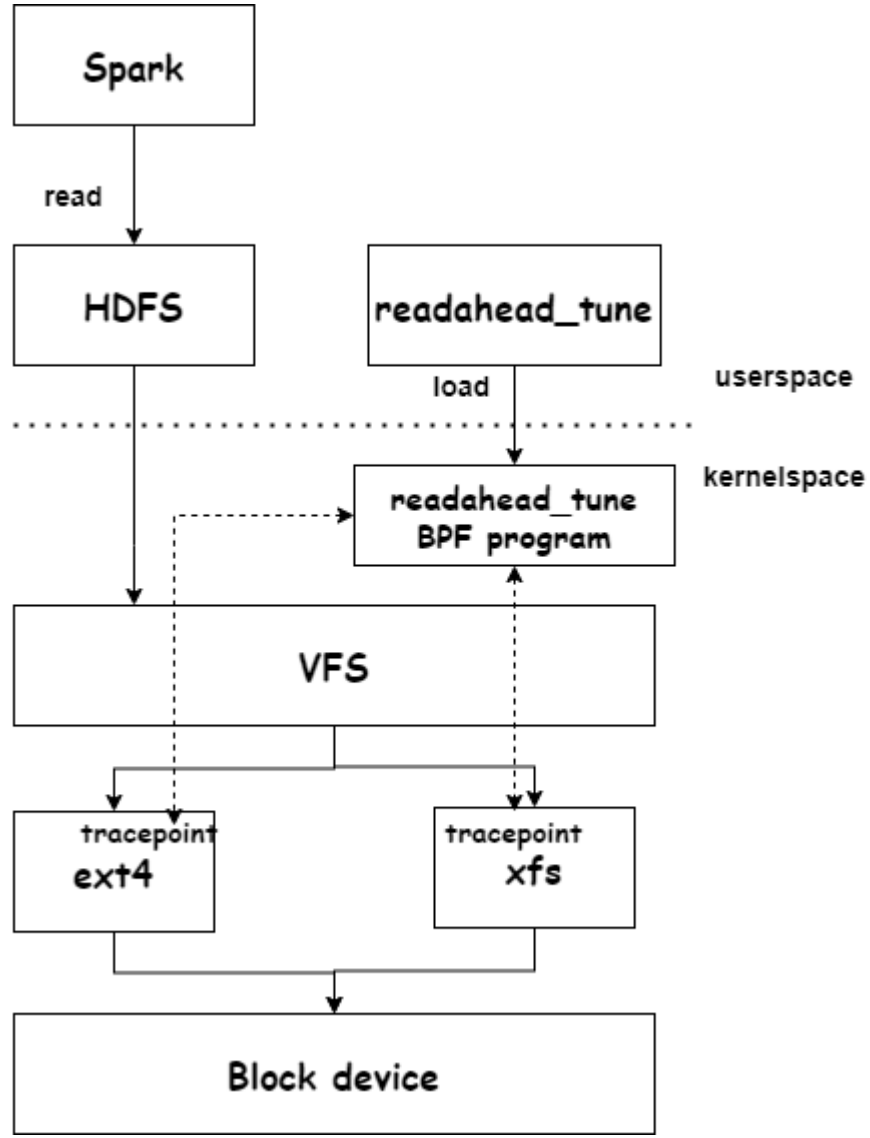
Case Study – Spark file read(optimization)

优化思路：

- 通过blk_文件名前缀过滤出spark所读的元数据文件、块数据文件
- 根据文件大小阈值(filesz)，对较小文件的读操作设置文件预读
- 跟踪文件读操作行为（读次数、顺序读次数等），以最近时间窗口的spark文件读行为模式，进行文件读操作设置：
  - 顺序读比例超过阈值，清除随机读模式
  - 顺序读比例低于阈值，设置随机读模式

主要工作：

- 内核：需要提供可读写的跟踪点用于跟踪xfs/ext4文件系统的读操作，并提供了和追踪文件读行为的BPF program的双向通信的能力
- BPF program追踪xfs/ext4 spark场景的读操作，根据spark文件的读行为调整spark文件读模式
- BPF控制程序(control process)，加载/attach BPF程序至内核，并在后台运行



openEuler

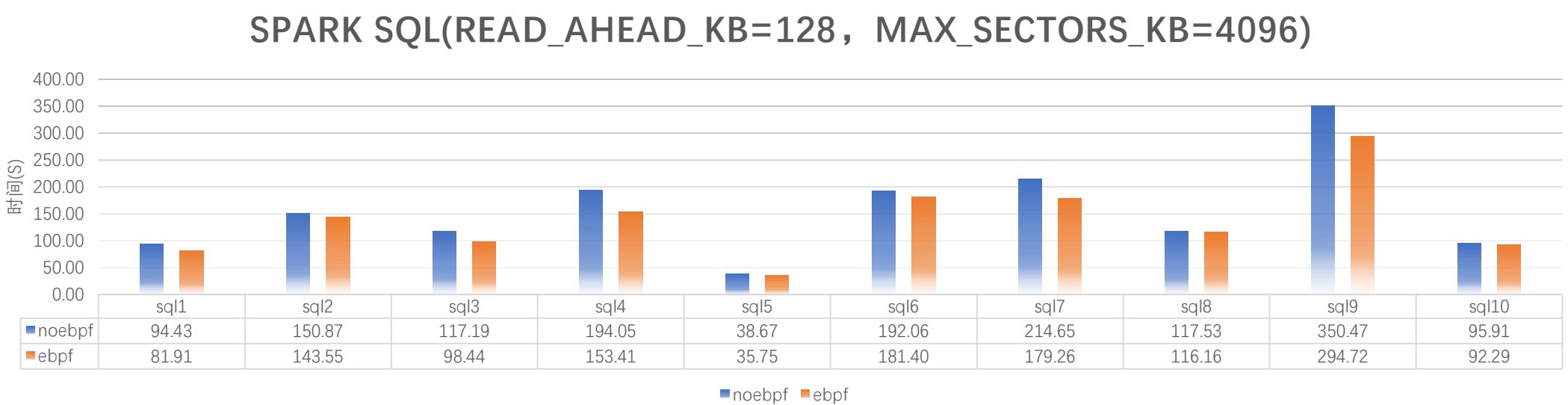
Case Study – Spark file read(Result)

测试结果：

- sql1 ~ sql10表示不同特征类型的I/O负载
- 借助ebpf使能策略，性能优化约**10%**

核心参数：

- https://gitee.com/openeuler/A-Tune-BPF-Collection/blob/master/readahead_tune.conf
- filesz-threshold：低于文件大小阈值的文件读操作都设置预读
- total-read-threshold：采样文件读操作的总次数阈值，超过该阈值，进行新一轮采样
- read-time-threshold：采样文件读操作的时间阈值，超过该阈值，进行新一轮采样
- lower-bound-percentage：采样周期内，顺序读比例的下限，低于该阈值，设置随机读
- upper-bound-percentage=：采样周期内，顺序读比例的下限，低于该阈值，清除随机读



	sql1	sql2	sql3	sql4	sql5	sql6	sql7	sql8	sql9	sql10
noebpf	94.43	150.87	117.19	194.05	38.67	192.06	214.65	117.53	350.47	95.91
ebpf	81.91	143.55	98.44	153.41	35.75	181.40	179.26	116.16	294.72	92.29

openEuler

Road Map for A-Tune BPF collection

1. ***收集内核启发式场景的调优策略，利用BPF程序进行定制***
2. A-Tune + BPF program：利用A-Tune调整BPF程序的关键参数
3. 根据A-Tune的调优数据，能够定制化BPF程序策略以替换内核启发式策略

openEuler