阿里云 | 奥运会全球指定云服务商

CFS调度器性能评价

陈善佩 | 阿里云-操作系统团队

2019.10.20

背景

阿里云 | 奥运会全球指定云服务商

- CFS调度策略越来越复杂，但缺乏系统的性能评价方法
  - 社区：
    - 性能评测方法简单，以少数几个benchmark来衡量性能改进，经常性的出现性能回退
    - 90001d67 sched/fair: Fix wake_affine() for !NUMA_BALANCING　　导致大量应用性能出现明显回退
    - 678d5718 sched/fair: Optimize cgroup pick_next_task_fair()　　导致大量应用性能波动巨大
  - 企业：
    - 缺乏可信性能数据来选择稳定可靠的内核版本
    - 难以评价自研或backport patch的性能好坏
    - 常常导致内核团队在调度子系统性方向发展缓慢

目标

- 建立调度器性能评价方法
  - 可信的benchmark集合
    - 基本操作测试：unix-bench, will-it-scale, lmbench
      - 调度器相关的测试用例中用到，主要用于判断性能变化是不是由基本操作引起的
    - 调度器相关测试：
      - 黑盒测试：hackbench, sysbench, tbench …
      - 白盒测试：缺少调度器相关的白盒测试用例
  - 综合性的评分机制
  - 采集历史版本的性能数据，建立性能数据库，分析性能变化趋势

白盒测试

- 将调度器按功能划分，为每个子项开发测试用例
  - **公平性**：运行时间上的公平
    - 任务权重 相同/不同
    - cpu.shares 相同/不同
    - 单cpu和多cpu的场景
    - 负载程度的差异
  - **优先级**：优先级高，调度延迟低
    - cpu.shares不同
    - 任务权重不同
    - 任务类型不同：经常睡眠任务 vs CPU密集型任务
  - **CPU利用率**
    - Load balance
    - sched overhead
    - cpu quota

综合评分机制

- **三个问题**
  - 在综合打分时，我们常常会想到按权重累加，那如何分配权重呢？
  - test cases的结果取值范围差异明显，存在数量级上的差异，如何消除这种差异？
  - 有些test case的结果越大越好（吞吐），有些越小越好（延迟），如何调和这种矛盾？

综合评分机制

- **问题一：**如何分配权重

  - 在分配过程中，我们进一步将调度器直接相关的test case分成两类：

    综合性：测试调度器整体性能

    局部性：测试调度器局部性能

  - 权重分配原则：综合性test cases权重要高于局部性cases

  - 不同业务场景：各个case权重不一致

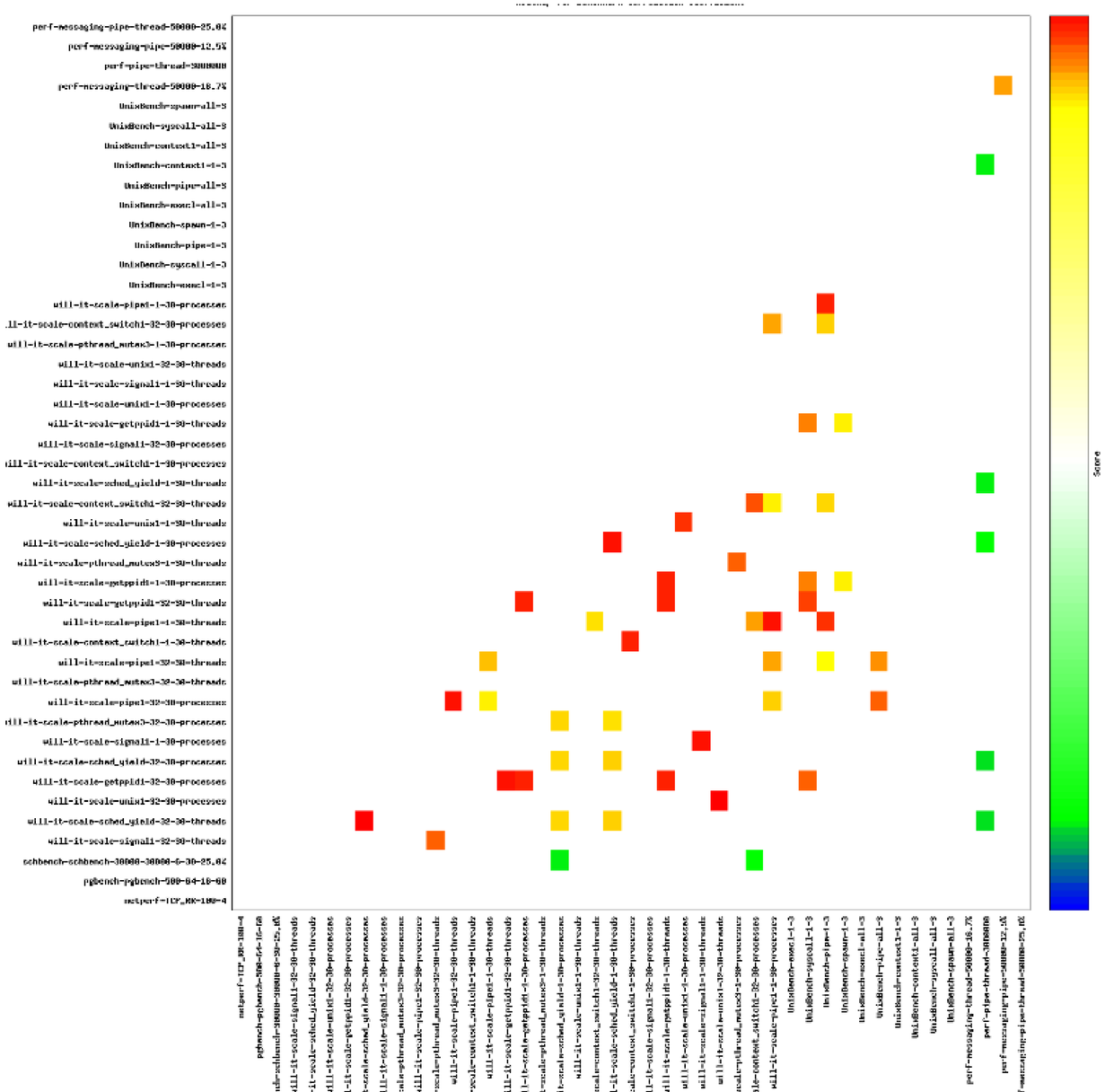
综合评分机制

- **问题二：**如何消除数值取值范围的差异
  - 数学变换：“中位数 + 对数 ”变换

1. 基于各个commit的历史性能数据，统计某个test cases结果的中位数；
2. 将测试结果除以对应的中位数，得到比例 R；
3. 如果各个case的 R 数量级仍然相差很大，则再做log变换：L = log(R)；
4. 按权重累加各个test case的分数 有 S = c1*L1 + c2*L2 + c3*L3 ... ，其中Sum（c） = 1
5. 线性变换 score = 100 + S * 100，放大差异

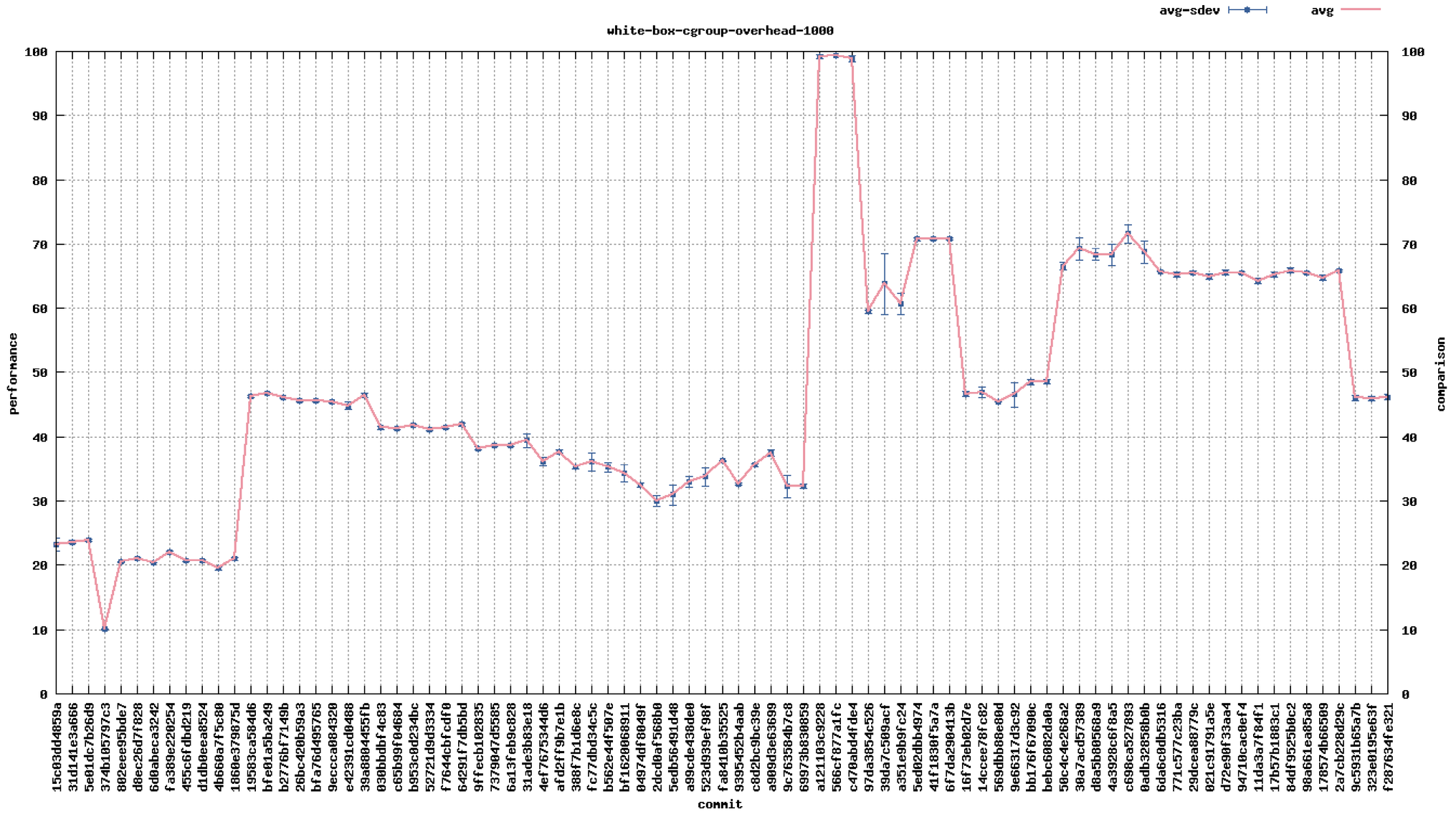
- **问题三：**如何调和性能指标性质不一致
  - 对于数值越小越好的test case，将“问题二”第2步的 R 取倒数，R = 1 / R。

数据分析工具：相关系数


perf-messaging-pipe-thread-50000-25.8%
perf-messaging-pipe-50000-12.5%
perf-pipe-thread-300000
perf-messaging-thread-50000-18.7%
UnixBench-spawn-all-3
UnixBench-syscall-all-3
UnixBench-context1-all-3
UnixBench-context1-1-3
UnixBench-pipe-all-3
UnixBench-execl-all-3
UnixBench-spawn-1-3
UnixBench-pipe-1-3
UnixBench-syscall-1-3
UnixBench-execl-1-3
will-it-scale-pipe1-1-30-processes
will-it-scale-context_switch1-32-30-processes
will-it-scale-pthread_mutex3-1-30-processes
will-it-scale-unix1-32-30-threads
will-it-scale-signal1-1-30-threads
will-it-scale-unix1-1-30-processes
will-it-scale-getppid1-1-30-threads
will-it-scale-signal1-32-30-processes
will-it-scale-context_switch1-1-30-processes
will-it-scale-sched_yield-1-30-threads
will-it-scale-context_switch1-32-30-threads
will-it-scale-unix1-1-30-threads
will-it-scale-sched_yield-1-30-processes
will-it-scale-pthread_mutex3-1-30-threads
will-it-scale-getppid1-1-30-processes
will-it-scale-getppid1-32-30-threads
will-it-scale-pipe1-1-30-threads
will-it-scale-context_switch1-1-30-threads
will-it-scale-pipe1-32-30-threads
will-it-scale-pthread_mutex3-32-30-threads
will-it-scale-pipe1-32-30-processes
will-it-scale-pthread_mutex3-32-30-processes
will-it-scale-signal1-1-30-processes
will-it-scale-sched_yield-32-30-processes
will-it-scale-getppid1-32-30-processes
will-it-scale-unix1-32-30-processes
will-it-scale-sched_yield-32-30-threads
will-it-scale-signal1-32-30-threads
schbench-schbench-30000-30000-6-30-25.8%
pgbench-pgbench-500-64-16-60
netperf-TCP_RR-100-4
Score

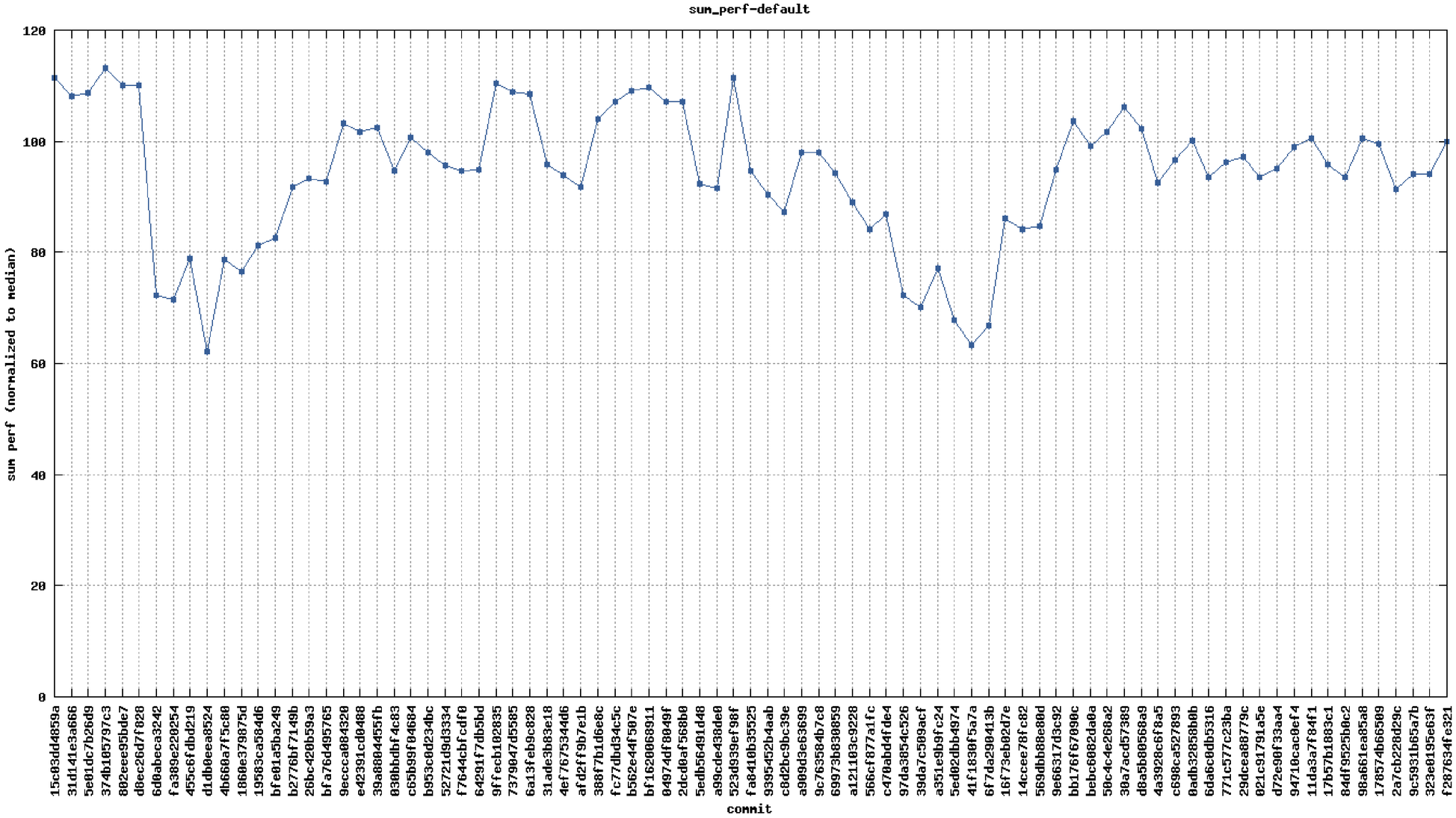
数据分析工具：单项性能

阿里云 | 奥运会全球指定云服务商



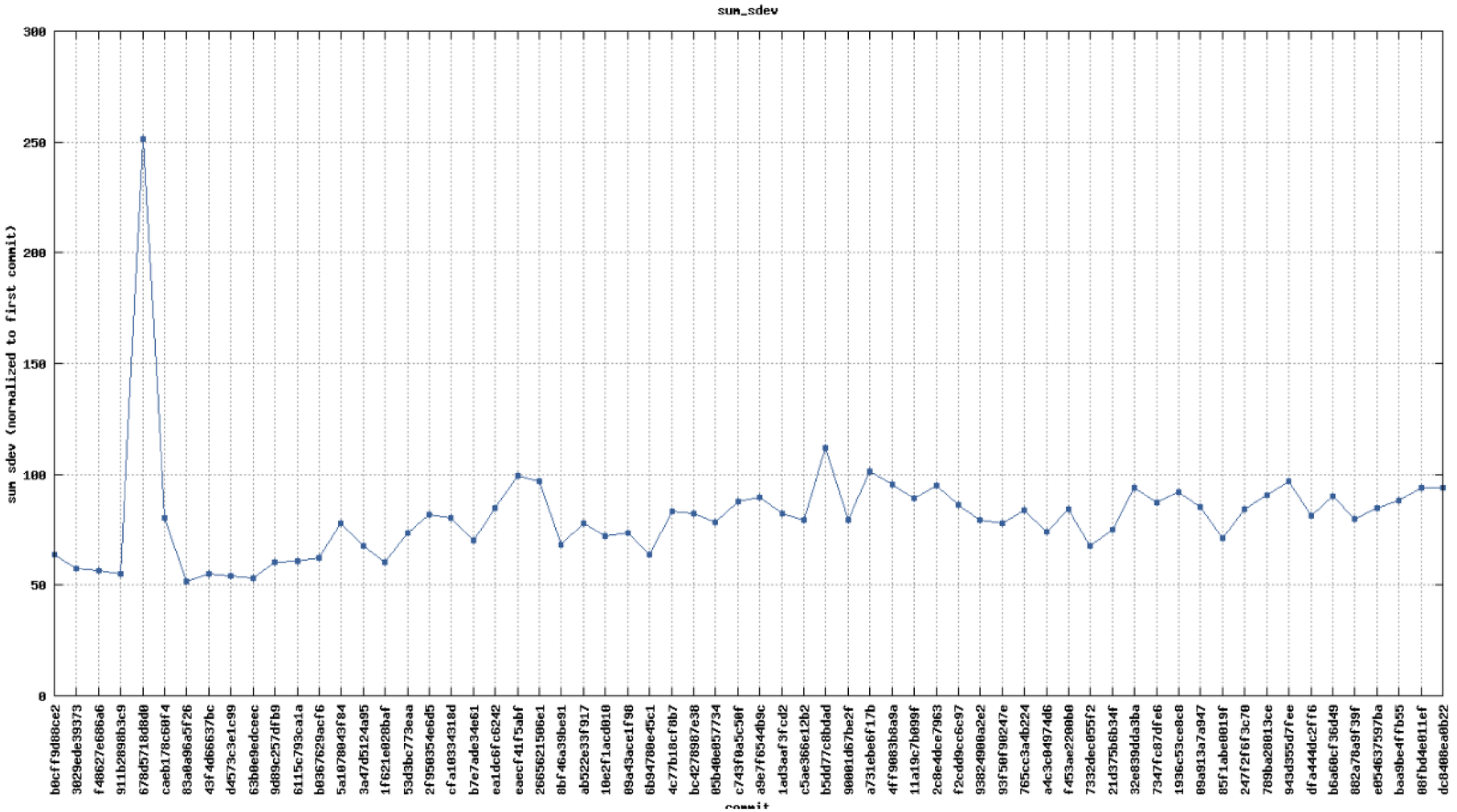
数据分析工具：综合性能

阿里云 | 奥运会全球指定云服务商



数据分析工具：累加波动

阿里云 | 奥运会全球指定云服务商



奥运会全球指定云服务商