Whamcloud

如何有效地报告Lustre文件系统的缺陷

杨　升

DDN® STORAGE

Lustre特点

- Lustre作为模块运行在kernel中
- 增加了跟踪和维护的难度
- 缺陷总是不可避免的
- 基本信息是必需的
- 有效的信息有助于解决问题
- 过多的无效信息会降低效率
- 尽力使用标准工具搜集信息

缺陷现场

- 缺陷导致的后果
- 缺陷的现场信息
- 详细的现场信息是解决问题的关键
- 过程信息和数据信息
- 现场有时效性
- 缺陷的后果和现场在时间上未必重合
- 后果和现场之间不一定有清晰的联系
- 可重现和不可重现

常见缺陷类型

Whamcloud

- SoftLockup
- OOM
- Crash
- Lustre Log
- 性能问题

SoftLockup

Whamcloud

- 现场信息一般能够保留
- 有足够的时间搜集信息
- 可是很难及时引起注意
- 通常涉及到锁的问题
- 过程信息就很有用，例如stack trace
- 可能需要多个节点的信息

OOM

Whamcloud

- 内存占用问题较难在早期发现
- 注意搜集slabinfo & free & ps & /proc/meminfo
- 由于系统响应差，信息搜集困难
- 发生OOM后往往已经错过现场
- OOM本身可能会带来混乱
- 问题定位比较困难
- 不推荐drop_cache

Crash

Whamcloud

- crash时刻console的输出是关键
- 推荐部署console设备
- 可以得到vmcore
- 推荐配置kdump
- 最好能提供debuginfo package
- 可以得到缺陷后果的详细信息
- 某些情况下现场已不存在
- LBUG和kernel本身都会触发

Lustre Log

Whamcloud

- Lustre专有机制
- 事先部署记录点
- 可根据不同子系统和级别来输出信息
- -1 log非常有用，可是在生产环境中不现实
- 包括了过程和数据信息
- 在某些情况下对于解决Lustre本身的问题是唯一的办法
- 搜集之前适当扩大缓冲区
- 搜集的时机非常重要
- 长时间的搜集可以使用文件记录
- client & server 端的协调

性能问题

- ftrace
- Perf tools
- Flame Graphs
- Blktrace

Thank You!

再 见

DDN® STORAGE