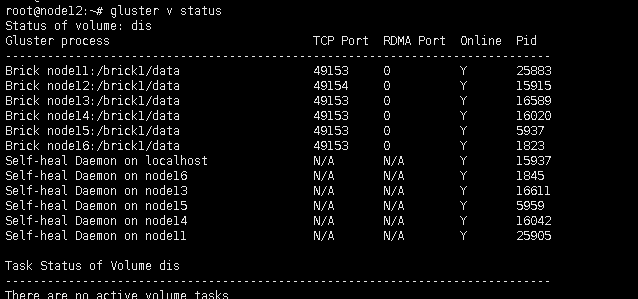
1. 硬件环境

|  |  |  |  |
| --- | --- | --- | --- |
| node11-node16 | Arm385 | server | 2核2G |
| Node120 | X86 | client | 32核96G |

1. 集群信息

|  |  |  |
| --- | --- | --- |
| 集群 | Node11-node16 | client |
| 卷 | 4：2纠删 | X86 |



1. 读写性能

测试命令iozone -i 0 -i 1 -r 512k -s 50g -t 4 -F file1 file2 file3 file4 总数量200G

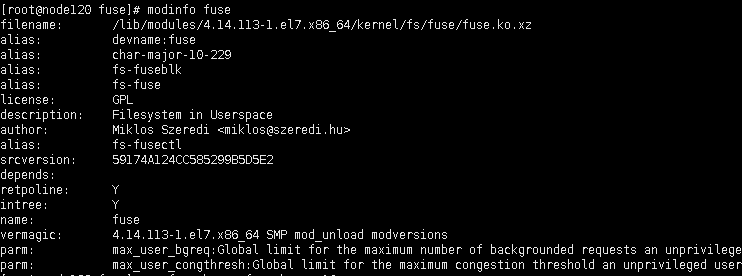
原生glus性能

|  |  |  |  |
| --- | --- | --- | --- |
| 模型 | io大小 | 带宽(MB/s) | 网络流量MB/s |
| 写100% | 512k | 287 | 420（多纠删码的流量） |
| 读100% | 512k | 234 | 234 |

调整iobuf后读写性能

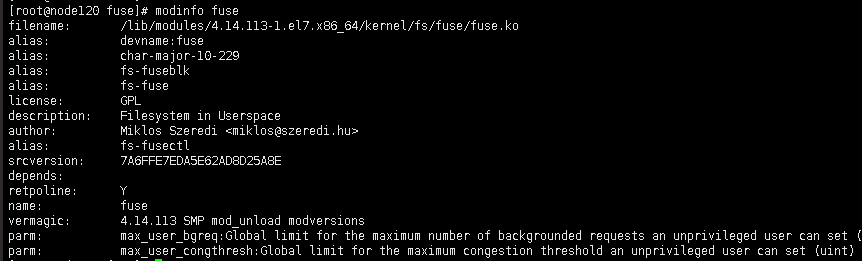
|  |  |  |  |
| --- | --- | --- | --- |
| 模型 | io大小 | 带宽 | 网络流量MB/s |
| 写100% | 512k | 428MB | 630（多纠删码的流量） |
| 读100% | 512k | 283 | 360 |

1. 原生fuse kernel信息



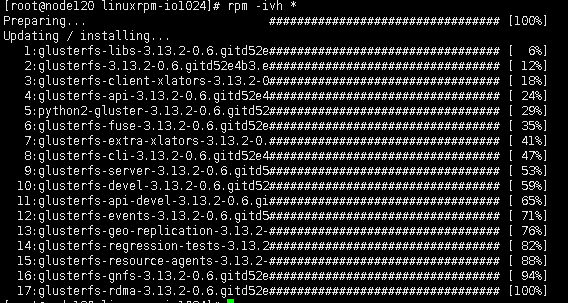


1. 修改kernel重新加载fuse

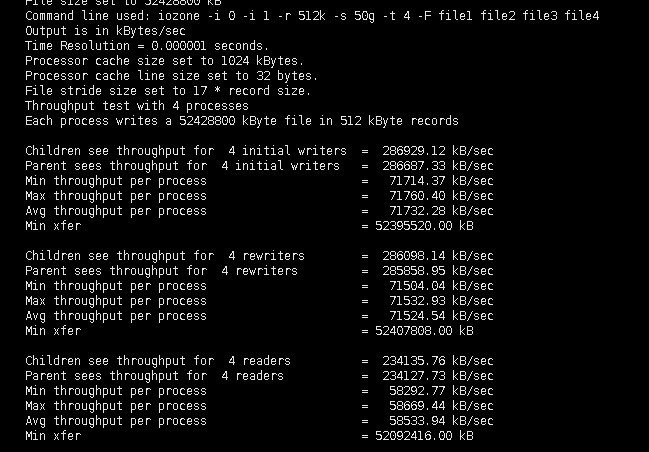


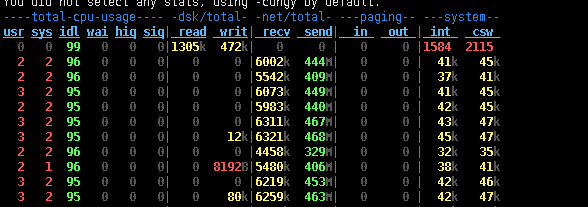


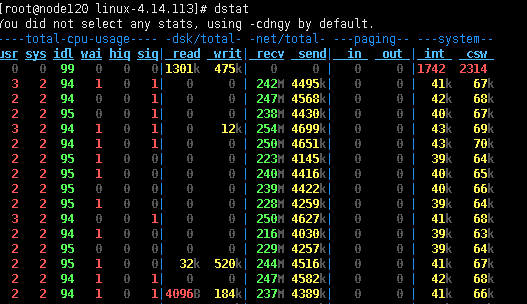
1. 安装修改iobuf后版本



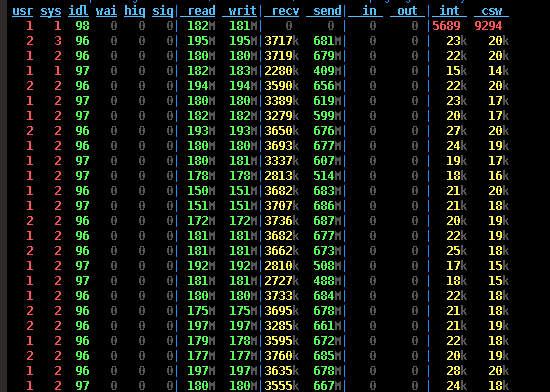
1. 原生glus挂载读写测试截图

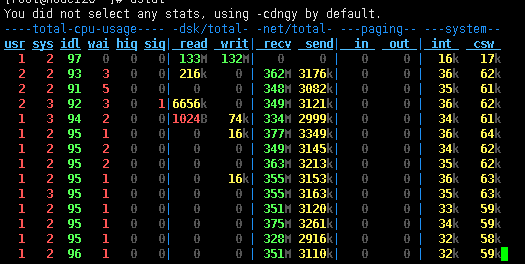






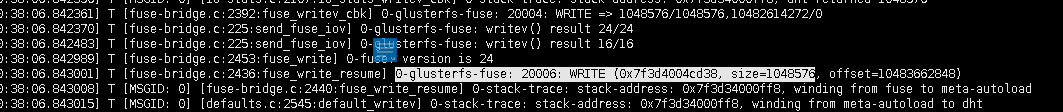
1. 修改iobuf后glus挂载读写测试截图



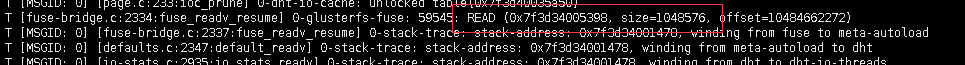


1. Iobuf验证

写



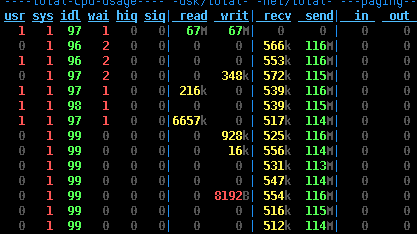
读



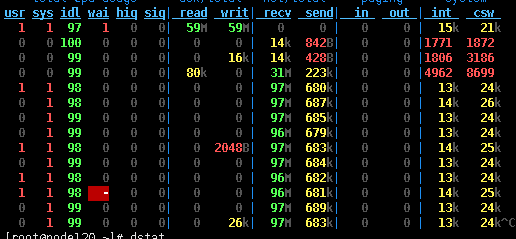
1. 测试io重读现象

为了清晰说明问题，直接使用dht单卷来分析（即非纠删非副本），arm单节点是千兆网口

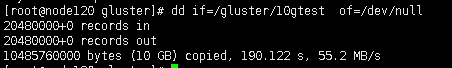
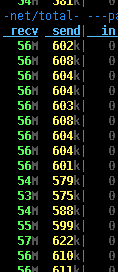
|  |  |  |  |
| --- | --- | --- | --- |
| 模型 | io大小 | 带宽 | 网络流量MB/s |
| 写100% | 1M | 115 | 115 |
| 读100% | 1M | 61 | 97 |
| 原iobuf读 | 1M | 55 | 55 |









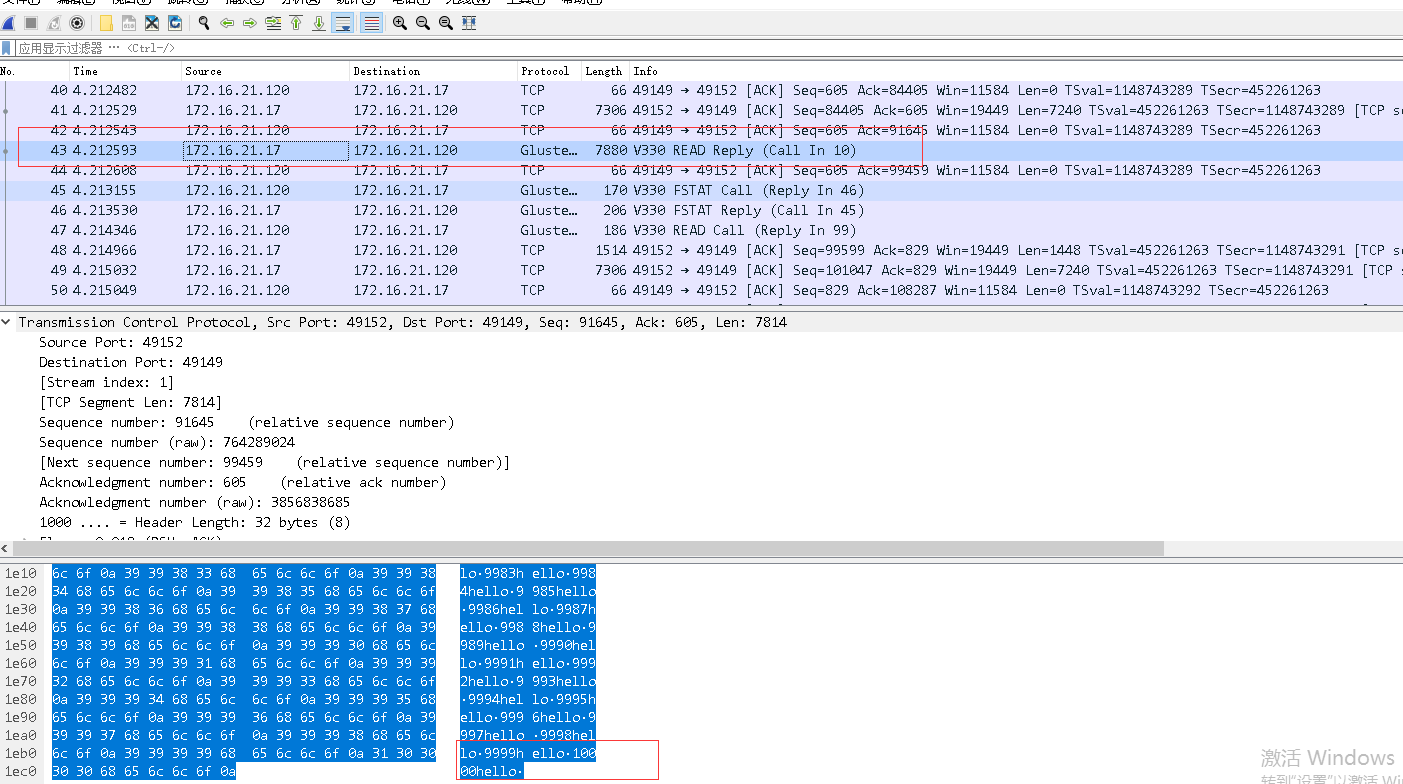


1. 准备1个内容为1hello ~ 10000hello的文件，直接读取内容，并tcpdump抓取数据流量

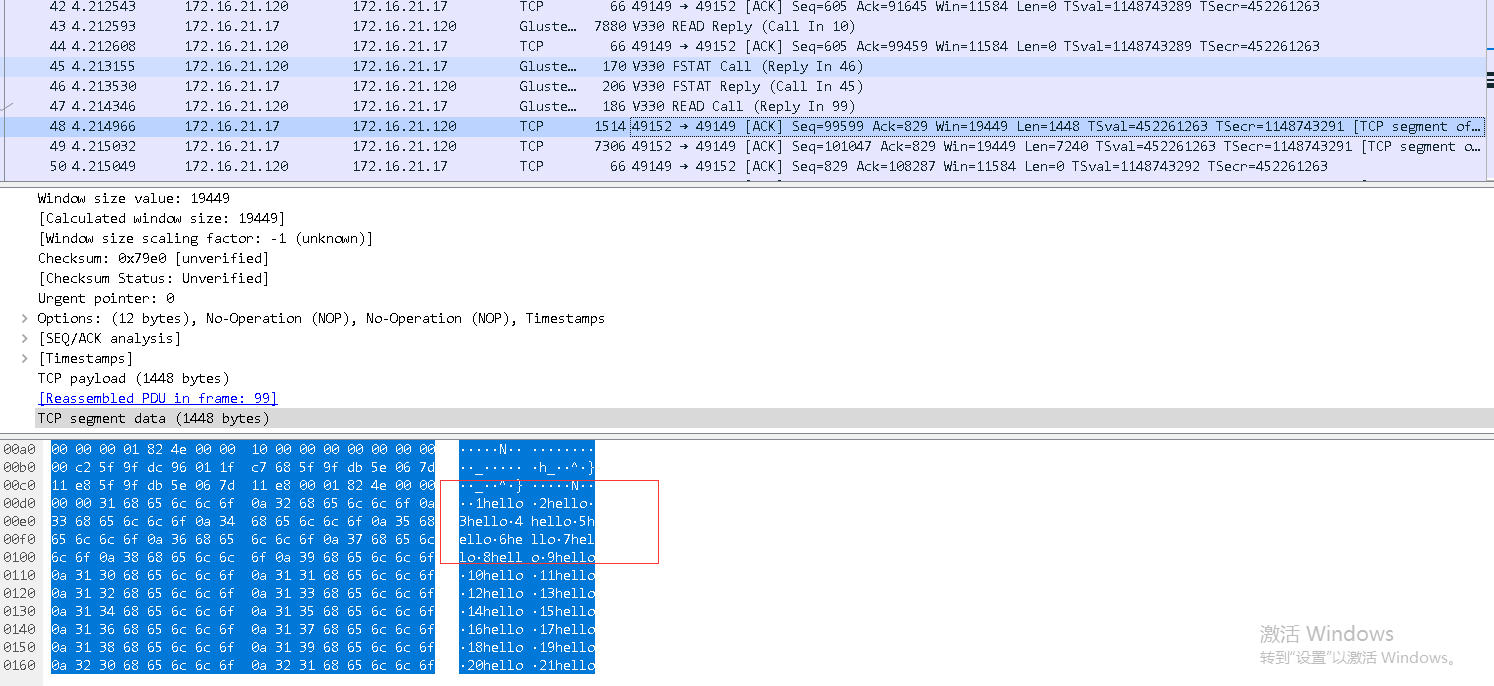
通过解析抓包文件内容可以看到在43行处17节点已经向120节点发送了最后的数据内容(...10000hello)，接下来48行开始继续重头读文件内容(1hello...)，最终在99行第二次读完了整个数据(...

10000hello).

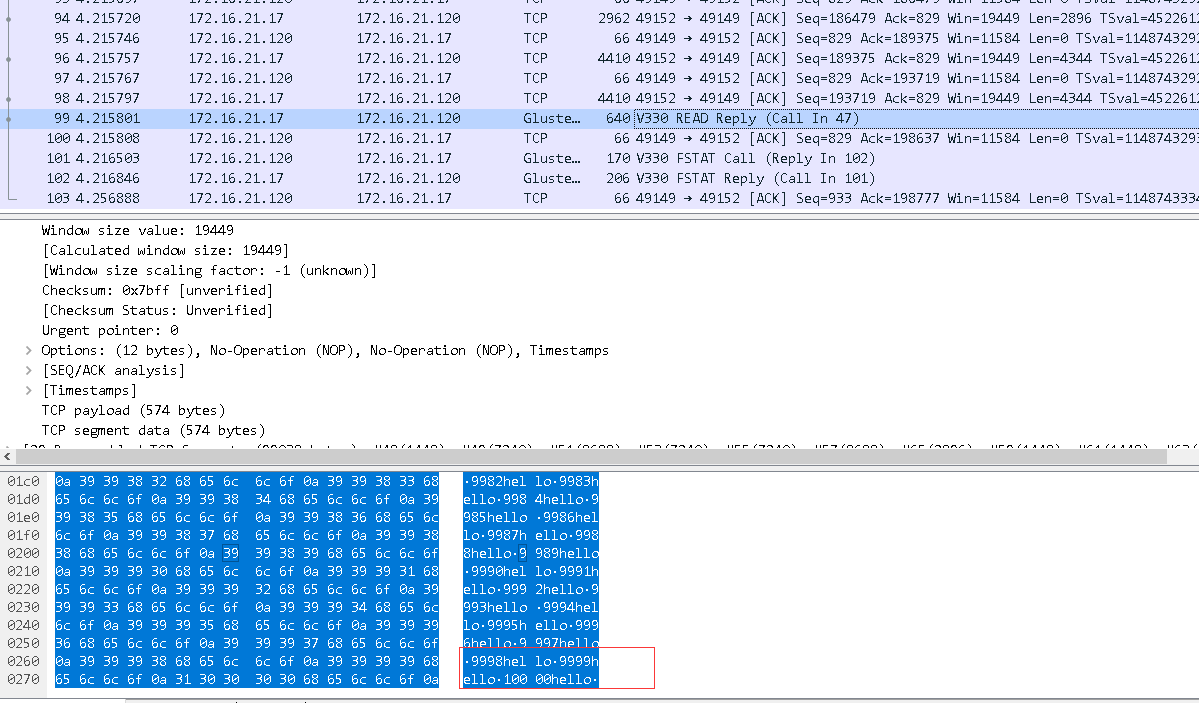
第一次读完数据

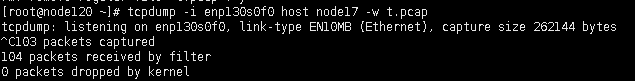


第二次读



最终第二次读完了数据







1. 结论

调整iobuf后对集群性能有提升1.5倍以上，但会已经重读的现象