Lustre Metadata Performance Testing Best Practice

Cheng Shao cheng_shao@xyratex.com



Outline

- Lustre metadata performance test overview
- Test methodology
- Lustre Metadata performance issue diagnosis



Overview

- Lustre metadata performance was less well-studied than data performance in the past
- More attentions on metadata performance lately
 - HPC applications generating huge number of tmp files
 - New application models
 - auditing
 - disaster recovery
 - NFS/CIFS server
 - Competition with other distributed/parallel file system and data storage solutions



- Metadata performance benchmarking tools
 - -mdtest
 - -mds-survey
 - mdsrate
 - metabench
 - -bonnie++
 - SPECsfs
 - other home grown tools



- MDS Hardware Configuration

 backend storage
 - –ldiskfs journal
 - CPU and memory
 - -networks (LST)







- mdtest specifics
 - number of tasks
 - -number of files to eliminate caching effect
 - multiple iterations to eliminate outliers
 - shared mount vs. unique mount
 - '-a' and '-o' option to exclude OST impact
 - directory structure



- Test Design
 - -increase number of tasks per client node
 - increase number of client nodes while keeping tasks per node constant
 - monitoring and profiling during the test
 - -list test factors and change one factor at a time
 - -try to give possible explanation to the difference
 - -testing with non-empty filesystems



- Issue types
 - -performance degradation
 - -fail to meet certain performance target
 - end-user complains
- Diagnosis methods
 - -profiling
 - -Lustre stats
 - crash dump



- Profiling tools
 - load
 - iostat
 - perf / oprofile
 - -latency
 - latencytop
 - systemtap



- Lustre stats
 - -stats
 - -req_history
 - -timeouts
 - -ldlm stats



apacity in us

0.98

0.96

0.38

0.66

0.00

0.00

0.00

0.00

0.00

0.00

0.00 0.00

0.00 0.00

MDT Perf	formand	e Data				OST Perfo	rma
File Configur	e						_
sangreal							
	angreal-M	DT0000 20 ⁴	13-03-0	8 18:16:45	.0		
%CPU	%	KB		%Inodes		OstName	Read
0.28		0.20		0.00			Rate
0.20		Camolo	0	0.00		sangreai-OS 10000	0.00
Operation	Samples	/Sec	Value	Std Dev	Units	sangreat-OST0001	0.00
close	0	0.00	0.00	0.00	regs	sangreal-OST0002	0.00
connect	0	0.00	0.00	0.00	regs	sangreat-0510005	0.00
create	0	0.00	0.00	0.00	regs	sangreal-OST0004	0.00
destroy	0	0.00	0.00	0.00	reqs	sangreal-OST0006	0.00
disconnect	0	0.00	0.00	0.00	reqs	sangreal-OST0007	0.00
getattr	0	0.00	0.00	0.00	reqs	AGGREGATE	0.00
getxattr	0	0.00	0.00	0.00	reqs	MAXIMUM	0.00
link	0	0.00	0.00	0.00	reqs	MINIMUM	0.00
llog_init	0	0.00	0.00	0.00	reqs	AVERAGE	0.00
mkdir	0	0.00	0.00	0.00	reqs		
mknod	0	0.00	0.00	0.00	reqs		
notify	0	0.00	0.00	0.00	reqs		
open	0	0.00	0.00	0.00	reqs		
process_config	0	0.00	0.00	0.00	reqs		
quotactl	0	0.00	0.00	0.00	reqs		
reconnect	0	0.00	0.00	0.00	reqs		
rename	0	0.00	0.00	0.00	reqs		
rmdir	0	0.00	0.00	0.00	reqs		
setattr	0	0.00	0.00	0.00	reqs		
statfs	0	0.00	0.00	0.00	reqs		
unlink	-	0.00	0.00	0.00	TO A		ALC: NO.

	mana		20 4000 30	Val NOCO			UN				
OLI	nanci	Storage Hardware	5U84 Disk A	rray		181	Installed				
		Storage Media	HDD	HDD		14760					
		SSD			360						
-		Power & Cooling	Power Supp	iy		360					
			Cooling Mod	lule		900	2				
_											
	Rate	2013 Xyratex Technolo	gy Limited A	I Rights Reser	ved.						
0	0.00	0.00	0.00	0.00	0.00			mytest204	0.00		
)1	0.00	0.00	0.00	0.00	0.00			mytest205	0.00		
2	0.00	0.00	0.00	0.00	0.00			AGGREGATE	0.00		
)3	0.00	0.00	0.00	0.00	0.00			MAXIMUM	0.00		
)4	0.00	0.00	0.00	0.00	0.00			MINIMUM	0.00		
5	0.00	0.00	0.00	0.00	0.00			AVERAGE	0.00		
6	0.00	0.00	0.00	0.00	0.00						
17	0.00	0.00	0.00	0.00	0.00						

0.00

0.00

0.00

0.00 0.00

0.00

0.00

0.00

0.00

0.00

0.00

0.00

0.00



xyratex

- One example
 - metadata performance degradation observed in Lustre
 2.3 clients, comparing with Lustre 1.8.8 client
 - benchmarking details: create files under a flat directory and then delete
 - creation rate is identical between 2.3 and 1.8.8 clients
 - -deletion rate of 2.3 clients is 1/3 of 1.8.8 clients



- Investigation
 - lucky enough to replicate in house
 - use systemtap to profile latencies of unlink call stack
 - -sample script

```
global times[100]
probe module("lustre").function("ll_unlink").return {
        times["ll_unlink"] <<< (gettimeofday_us()-@entry(gettimeofday_us()))</pre>
3
probe module("mdc").function("mdc_unlink").return {
        times["mdc_unlink"] <<< (gettimeofday_us()-@entry(gettimeofday_us()))</pre>
probe module("mdc").function("mdc_reint").return {
        times["mdc_reint"] <<< (gettimeofday_us()-@entry(gettimeofday_us()))</pre>
probe end {
        foreach (func in times) {
                printf("\n\n==== %s stat ====\n", func)
                printf("total samples: %d, min %d, max %d, avg %d\n",
                        @count(times[func]),
                        @min(times[func]),
                        @max(times[func]),
                        @avg(times[func]))
                print(@hist_linear(times[func], 0, 45000, 400))
        3
        printf("\nbye now\n")
```



- Investigation
 - comparing the profiling results of 1.8 and 2.3 clients
 - sample outputs (latency of ll_unlink)

1.8 clients

2.3 clients

total :	samples: 8000, min 211, max 144987, avg 506	
value		count
0	022222222222222222222222222222222222222	7989
2000	1 contraction of the second seco	6
4000	1 contraction of the second seco	0
6000	1	0
	~	
40000	1 contraction of the second seco	0
42000	1 contraction of the second seco	0
44000	1	2
46000	1 Contraction of the second seco	0
48000	The second se	0
	~	
54000	1	0

total s	amples: 8000, min 249, max 145504, avg 1539	
value		count
0	000000000000000000000000000000000000000	7391
2000	1 contract of the second se	30
4000	1 contract of the second se	47
6000	I	41
8000	I	47
10000	I	75
12000	I	37
14000	I construction of the second sec	83
16000	I contract the second se	104
18000	I contract the second se	69
20000	I construction of the second se	25
22000	1	19

 most part of the latency comes from clients waiting for responses from MDS



- Investigation
 - server side code path
 - mdt_reint_unlink
 - mdt_object_find_lock // lock parent dir
 - mdd_unlink // actual unlink
 - profiling results

function	with 1.8 clients	with 2.3 clients
mdt_reint_unlink	304	1328
mdd_unlink	75	76
mdt_object_find_lock	51	1110

 – why does it take longer to lock the shared parent directory with 2.3 clients



- Investigation
 - code review on the benchmark application.
 - -file creation work flow
 - -file unlink work flow
 - readdir to get next direntry
 - stat direntry to skip everything other than regular files
 - unlink the direntry
- Difference between Lustre 1.8 and 2.3
 - readdir chunk size gets increased from 1 page to 256 pages
 - -lock cancellation takes longer on the client side



- Conclusion
 - certain workflow in benchmarking tools or user applications may sometimes not be able to achieve the optimal metadata performance Lustre could offer
 - systemtap is proved to be an effective tool for latency profiling and performance diagnosis



Thank You

cheng_shao@xyratex.com

