



TECHNICAL SPECIFICATION



ScaleFlux CSD 2000 NVMe SSD For openGauss

测试报告

Version 1.0

最后修改: 3/16/2022

目录

1. 测试目的	3
2. 测试结论	3
2.1. SSD 基准测试 (FIO 测试)	3
2.2. SYSBENCH OLTP 测试结论	4
2.3. Benchmark SQL TPC-C 测试结论	6
3. 测试详情	7
3.1. 软硬件配置	7
3.1.1. 主机和操作系统	7
3.1.2. openGauss 参数	8
3.2. 测试用例设计	8
3.2.1. FIO 基准测试	8
3.2.2. SYSBENCH OLTP 基准测试	9
3.2.3. BenchmarkSQL TPC-C 测试	9
3.3. 测试数据明细	10
3.3.1. FIO 基准测试数据	10
3.3.2. sysbench oltp 测试数据	10
3.3.3. BenchmarkSQL TPC-C 测试数据	14
参考	15

1. 测试目的

与普通 NVMe SSD 相比，透明压缩是 ScaleFlux CSD 2000 独有的特性，在 NVMe SSD 盘内部实现了数据路径上的压缩。从理论上分析，压缩能够带来性能提升，减少写放大，延长使用寿命等。

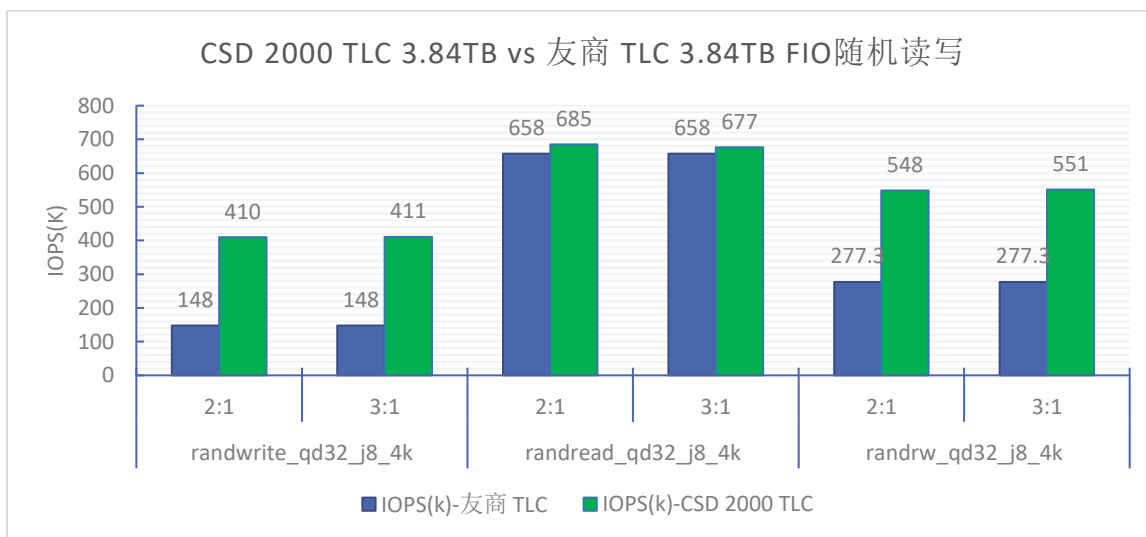
本测试目的是对 CSD 2000 的性能、功能做较全面的验证，并与友商 TLC 标盘做横向对比，以检验 CSD 2000 在 openGauss 上 TCO 下降和性能提升方面的收益。

2. 测试结论

通过 FIO、SYSBENCH OLTP 和 Benchmark SQL TPC-C 测试验证，当业务数据可压缩性很好时，CSD 2000 能节省数据存储的实际空间同时又能提升性能。跟 openGauss 结合使用时，通过降低 openGauss 的表的填充因子（Fill Factor）还可以进一步实现空间节省 2/3 和性能提升 60% 的双赢目标。

2.1. SSD 基准测试（FIO 测试）

- 在 2.9:1 压缩率下，相比友商 TLC，CSD 2000 TLC 在能够节省 65.5% 存储空间的同时，随机读 IOPS 提高了 2.89%，随机写 IOPS 提高了 177.7%，随机读写混合 IOPS 提高了 98%。

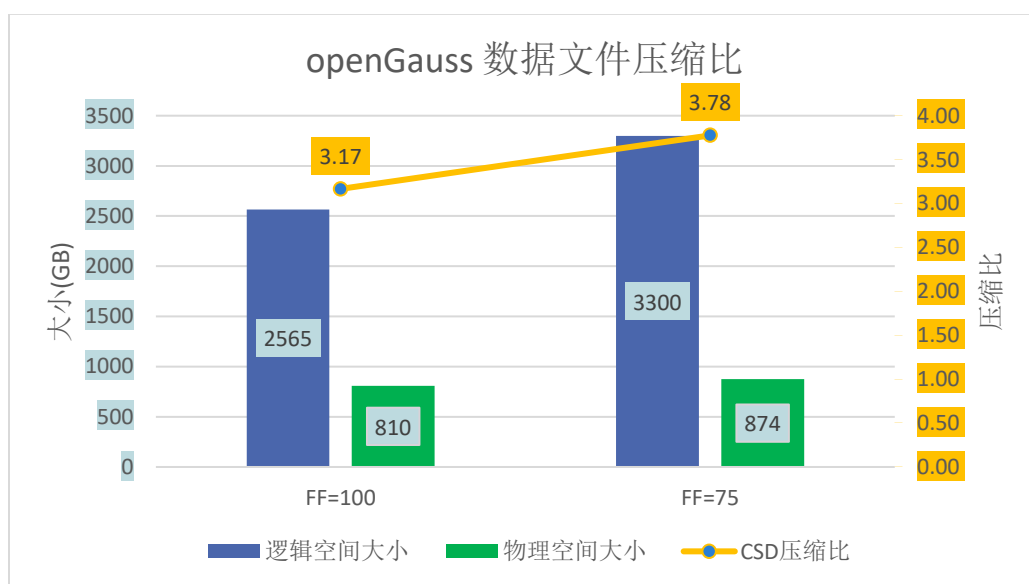


2.2. SYSBENCH OLTP 测试结论

SYSBENCH 测试数据是 100 表，每表 1 亿数据量。表的 Fill Factor 设置分别为 100 和 75。具体测试案例说明请参考：[SYSBENCH OLTP 基准测试](#)。

下面是不同 Fill Factor 设置下，openGauss 数据文件在 CSD 2000 上空间压缩比。

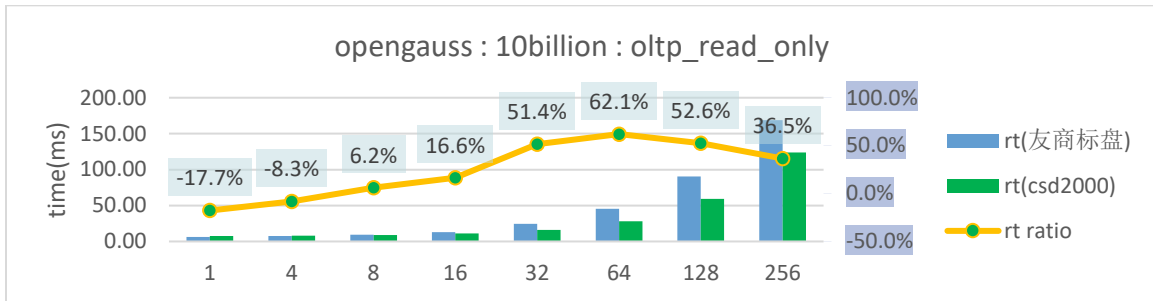
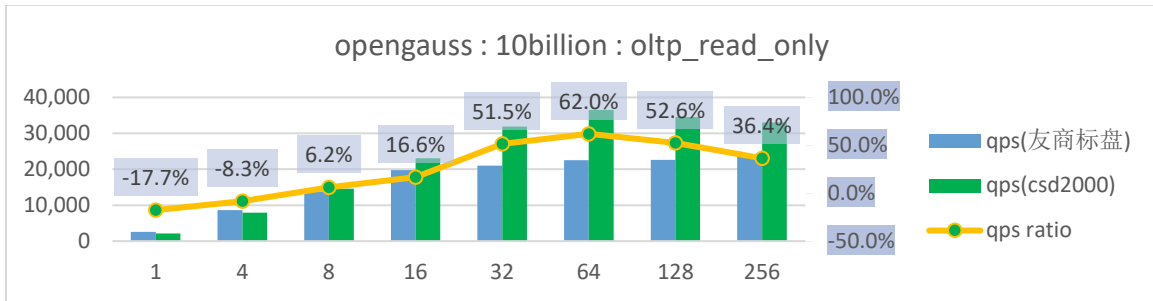
FillFactor	逻辑空间大小	物理空间大小	CSD 压缩比
FF=100	2565	810	3.17
FF=75	3300	874	3.78



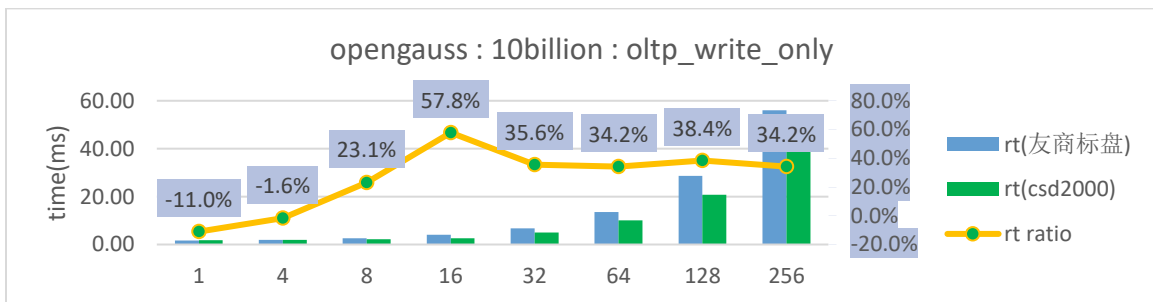
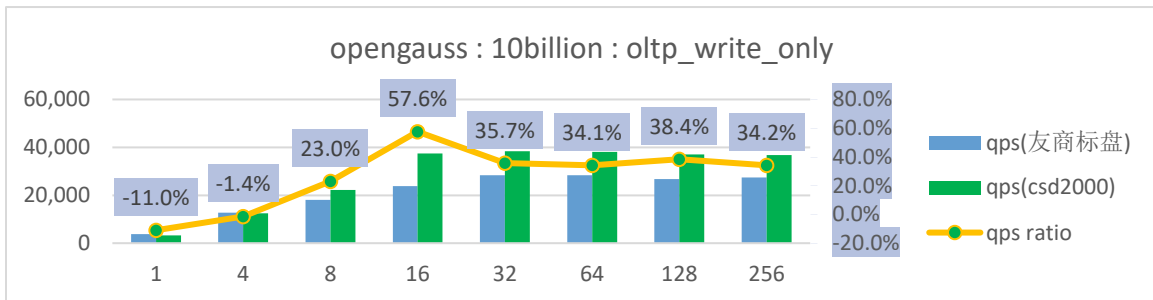
Fill Factor 使用默认值时，openGauss sysbench 数据在 CSD 2000 上压缩比有 3.16。继续降低 Fill Factor 值可以提升 openGauss 更新性能，数据库文件大小会从 2.6T 增长到 3.2T，但在 CSD 2000 上，实际大小只增长了 40GB 左右。数据最终在 CSD 2000 上压缩也增长到 3.77。

下面是 Fill Factor 在 75 的时候，openGauss 运行 sysbench 测试时，在友商标盘和 CSD 2000 上的性能对比图。可以看出当并发数达到 8 以后，CSD 2000 相比友商标盘，在读写方面性能都有不同程度的提升，最高提升到 75%。

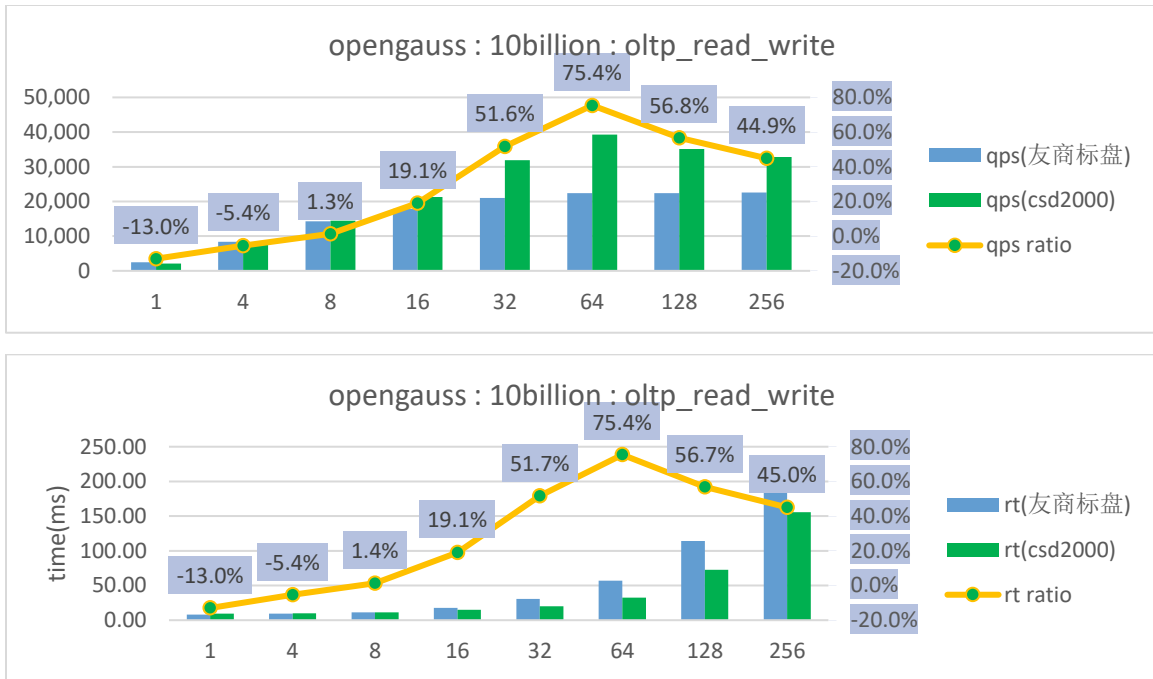
- 只读 read_only



■ 纯写 write_only



■ 读写混合 read_write



2.3. Benchmark SQL TPC-C 测试结论

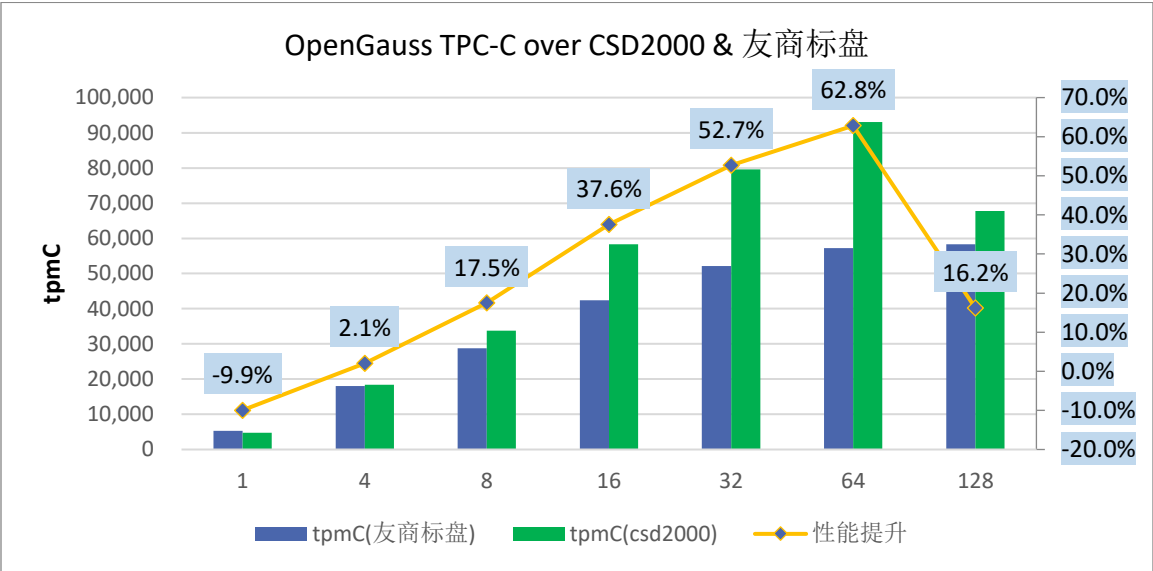
Benchmark SQL TPC-C 数据规模是 10000 仓。详细测试用例说明请参考：

[BenchmarkSQL TPC-C 测试](#)

将 openGauss 表的 Fill Factor 设置为 50 时，在标盘 SSD 下，openGauss 里这个数据库大小约为 1.7TB，在 CSD 2000 内部物理空间大小约为 600GB，CSD 压缩比约在 3.12。

空间的压缩，降低了 SSD 内部 GC 的频率，提升了高并发读写时性能，最大提升比例约 60%。

terminals	tpmC(友商标盘)	tpmC(CSD 2000)	性能提升
1	5,271	4,747	-9.9%
4	18,002	18,376	2.1%
8	28,731	33,757	17.5%
16	42,377	58,305	37.6%
32	52,096	79,569	52.7%
64	57,184	93,100	62.8%
128	58,327	67,775	16.2%



3. 测试详情

3.1. 软硬件配置

3.1.1. 主机和操作系统

具体项	当前型号
Linux 版本	CentOS 7.8
CPU 型号及其核心数	Intel(R) Xeon(R) Gold 6126 CPU @ 2.60GHz * 2
闪存卡型号和数量	1 * 友商 1 TLC 3.84TB vs 1 * CSD 2000 TLC 3.84TB
物理机厂商	Inspur
内核版本	3.10.0-693.el7.x86_64
数据库版本	openGauss v2.1
sysbench 版本	1.1.0
内存大小	128 GB, 32GB DIMM,2.6GHZ, Samsung
裸机/虚拟机/容器	物理机

3.1.2. openGauss 参数

数据库参数通过命令行设置，然后重启 openGauss 实例。

```
gs_guc reload -N all -I all -c "random_page_cost = '2'"
gs_guc reload -N all -I all -c "max_parallel_workers_per_gather = '0'"
gs_guc reload -N all -I all -c "maintenance_work_mem = '2GB'"
gs_guc reload -N all -I all -c "wal_compression = on" --
gs_guc reload -N all -I all -c "checkpoint_completion_target = 0.9"
gs_guc reload -N all -I all -c "min_wal_size = '1GB'" ---
gs_guc reload -N all -I all -c "max_wal_size = '48GB'" ---
gs_guc reload -N all -I all -c "wal_keep_segments = 1000"
gs_guc reload -N all -I all -c "checkpoint_timeout='20min'"
gs_guc reload -N all -I all -c "log_statement = ddl"
gs_guc reload -N all -I all -c "effective_cache_size='90GB'"
gs_guc reload -N all -I all -c "checkpoint_segments=128"
gs_guc reload -N all -I all -c "password_encryption_type=0"
gs_guc reload -N all -I all -c "shared_buffers='30GB'"
gs_guc reload -N all -I all -c "max_process_memory='90GB'"
gs_guc reload -N all -I all -c "maintenance_work_mem='10GB'"
gs_guc reload -N all -I all -c "enable_wdr_snapshot=on"
gs_guc reload -N all -I all -c "autovacuum_max_workers=10"
```

3.2. 测试用例设计

3.2.1. FIO 基准测试

- 使用 FIO 测试顺序写(write)、顺序读(read)、随机写(randwrite)、随机读(randread)和随机读写(randrw)混合。
- 数据压缩比控制在 2.9。

```
buffer_compress_chunk=4K
buffer_compress_percentage=75
```

- 总共运行 10h 以上。

示例 - 块设备上的 4k 随机写

```
fio --ioengine=libaio \
--randrepeat=0 \
--norandommap \
--thread \
--direct=1 \
--name=block_randwrite_4k \
--rw=randwrite \
--bs=4k \
--numjobs=2 \
--iodepth=64 \
--group_reporting \
--random_generator=tausworthe \
--filename=/dev/sfdv0n1 \
--buffer_compress_chunk=4k \
--buffer_compress_percentage=75 \
```



```
--output=sfdv0n1_randwrite_4k.log
```

3.2.2. SYSBENCH OLTP 基准测试

- 使用 SYSBENCH 测试场景纯读 (oltp_read_only.lua) 、纯写 (oltp_write_only.lua) 、读写混合 (oltp_read_write.lua) 。
- 数据规模：100 表 * 1 亿/表 。
- 数据压缩率调整：默认 sysbench 是随机数字和字符串，为了模拟业务数据，修改数据生成逻辑，允许字符串字段里有部分值重复。具体见下面说明。
- 表设置：fillfactor 设置为 75 。
- 分别创建两个 openGauss 实例，数据库文件分别在友商标盘和 CSD 2000 上。

备注：

- 调整 sysbench 数据生成逻辑，修改 oltp_common.lua 脚本。

```
-- Template strings of random digits with 11-digit groups separated by dashes

-- 10 groups, 119 characters
local c_value_template = "00000000000-00000000000-00000000000-" ..
    "00000000000-00000000000-00000000000-" ..
    "00000000000-#####-#####-" ..
    "#####-"
--local c_value_template = "#####-#####-#####-" ..
-- "#####-#####-#####-" ..
-- "00000000000-#####-#####-" ..
-- "#####-"

-- 5 groups, 59 characters
--local pad_value_template = "#####-#####-#####-" ..
-- "#####-#####-"
local pad_value_template = "00000000000-00000000000-00000000000-" ..
    "00000000000-#####"
```

3.2.3. BenchmarkSQL TPC-C 测试

- 使用开源的 BenchmarkSQL 测试，调整相应的并发数（terminals）观察每分钟订单创建数（tpmC）。
- 数据规模：10000 仓，最大表数据量在百亿级别。
- 表设置：fillfactor 设置为 75。
- 分别创建两个 openGauss 实例，数据库文件分别在友商标盘和 CSD 2000 上。

3.3. 测试数据明细

3.3.1. FIO 基准测试数据

■ FIO 测试数据整理

测试项	盘片型号	吞吐(MB/S)		IOPS(k)		IO 平均延迟(us)	
seq_write_128K_qd128_1job	友商 TLC	2899		22.1		5774	
	CSD 2000 TLC	2370		18.1		7054	
seq_read_128K_qd128_1job	友商 TLC	3139		23.9		5343	
	CSD 2000 TLC	3115		23.8		5385	
randwrite_qd32_j8_4k	友商 TLC	606		148		1728	
	CSD 2000 TLC	1685		411		620	
randrw_qd32_j8_4k	友商 TLC	796	341	194	83.3	699	1440
	CSD 2000 TLC	1581	678	386	165	502	372
randread_qd32_j8_4k	友商 TLC	2693		658		388	
	CSD 2000 TLC	2774		677		377	

3.3.2. sysbench oltp 测试数据

■ CSD 2000 压缩比数据

观察 openGauss 数据库文件所在 CSD 磁盘的压缩比， 3.16 。

```
[root@sfx111188 ~]# sh csd_size.sh sfdv1n1
Time - 2022-02-24-07:13:07
Device - /dev/sfdv1n1
Total capacity - 3577 GiB
Used space - 810 GiB
Free space - 2767 GiB
Logical data size - 2565 GiB
Compression ratio - 3.16 (logical data size / used space)
```

调整 Fill Factor 为 75 后，再运行 sysbench 测试 3 小时，压缩比稳定在 3.77 。

```
[root@sfx111188 ~]# sh ~/csd_size.sh sfdv1n1
Time - 2022-02-28-07:32:35
Device - /dev/sfdv1n1
Total capacity - 3577 GiB
Used space - 874 GiB
Free space - 2703 GiB
Logical data size - 3300 GiB
Compression ratio - 3.77 (logical data size / used space)
```

- openGauss 数据文件压缩比数据

观察 openGauss 的数据文件中表文件压缩比，抽样看一部分。压缩比在 3.19 左右。

```
[omm@sfx111188 24631]$ sudo sfx-filesize 42874* |grep .
Logical      Alocated      Physical      Ratio File
1073741824   1073745920    336293680    3.19 42874
Logical      Alocated      Physical      Ratio File
1073741824   1073745920    336718656    3.19 42874.1
Logical      Alocated      Physical      Ratio File
1073741824   1073745920    336827488    3.19 42874.10
Logical      Alocated      Physical      Ratio File
1073741824   1073745920    336957936    3.19 42874.11
Logical      Alocated      Physical      Ratio File
1073741824   1073745920    336710832    3.19 42874.12
Logical      Alocated      Physical      Ratio File
1073741824   1073745920    337108080    3.19 42874.13
Logical      Alocated      Physical      Ratio File
1073741824   1073745920    336864304    3.19 42874.14
Logical      Alocated      Physical      Ratio File
1073741824   1073745920    336852160    3.19 42874.15
Logical      Alocated      Physical      Ratio File
1073741824   1073745920    336819056    3.19 42874.16
Logical      Alocated      Physical      Ratio File
1073741824   1073745920    336747792    3.19 42874.17
Logical      Alocated      Physical      Ratio File
1073741824   1073745920    336961808    3.19 42874.18
Logical      Alocated      Physical      Ratio File
1073741824   1073745920    336687648    3.19 42874.19
Logical      Alocated      Physical      Ratio File
1073741824   1073745920    336523232    3.19 42874.2
Logical      Alocated      Physical      Ratio File
665706496    665710592     208809760    3.19 42874.20
Logical      Alocated      Physical      Ratio File
1073741824   1073745920    336428064    3.19 42874.3
Logical      Alocated      Physical      Ratio File
1073741824   1073745920    336581088    3.19 42874.4
Logical      Alocated      Physical      Ratio File
1073741824   1073745920    336741856    3.19 42874.5
Logical      Alocated      Physical      Ratio File
1073741824   1073745920    336942288    3.19 42874.6
Logical      Alocated      Physical      Ratio File
1073741824   1073745920    336704704    3.19 42874.7
Logical      Alocated      Physical      Ratio File
1073741824   1073745920    336929568    3.19 42874.8
Logical      Alocated      Physical      Ratio File
1073741824   1073745920    336542112    3.19 42874.9
Logical      Alocated      Physical      Ratio File
5464064      5468160       202768       26.95 42874_fsm
[omm@sfx111188 24631]$
```

- openGauss 事务日志 xlog 文件压缩比数据

观察 openGauss 的事务日志 xlog 文件压缩比, 抽样一部分, 压缩比在 2.26~2.56 左右。

```
[omm@sfx111188 gauss_xlog]$ sudo sfx-filesize 000000010000058A0000005* |grep .
```

	Logical	Alocated	Physical	Ratio	File
	16777216	16777216	6570192	2.55	000000010000058A00000050
	Logical	Alocated	Physical	Ratio	File
	16777216	16777216	6559088	2.56	000000010000058A00000051
	Logical	Alocated	Physical	Ratio	File
	16777216	16777216	6557280	2.56	000000010000058A00000052
	Logical	Alocated	Physical	Ratio	File
	16777216	16777216	6562032	2.56	000000010000058A00000053
	Logical	Alocated	Physical	Ratio	File
	16777216	16777216	6565536	2.56	000000010000058A00000054
	Logical	Alocated	Physical	Ratio	File
	16777216	16777216	6569664	2.55	000000010000058A00000055
	Logical	Alocated	Physical	Ratio	File
	16777216	16777216	5592912	3.00	000000010000058A00000056
	Logical	Alocated	Physical	Ratio	File
	16777216	16777216	7038576	2.38	000000010000058A00000057
	Logical	Alocated	Physical	Ratio	File
	16777216	16777216	7396864	2.27	000000010000058A00000058
	Logical	Alocated	Physical	Ratio	File
	16777216	16777216	7417616	2.26	000000010000058A00000059
	Logical	Alocated	Physical	Ratio	File
	16777216	16777216	7453872	2.25	000000010000058A0000005A
	Logical	Alocated	Physical	Ratio	File
	16777216	16777216	7323056	2.29	000000010000058A0000005B
	Logical	Alocated	Physical	Ratio	File
	16777216	16777216	7411424	2.26	000000010000058A0000005C
	Logical	Alocated	Physical	Ratio	File
	16777216	16777216	7479952	2.24	000000010000058A0000005D
	Logical	Alocated	Physical	Ratio	File
	16777216	16777216	7412096	2.26	000000010000058A0000005E
	Logical	Alocated	Physical	Ratio	File
	16777216	16777216	7424816	2.26	000000010000058A0000005F

▪ sysbench 表大小 (节选)

Fill Factor=100

```
--查出表大小按大小排序并分离 data 与 index
SELECT
table_name,
pg_size_pretty(table_size) AS table_size,
pg_size_pretty(indexes_size) AS indexes_size,
pg_size_pretty(total_size) AS total_size
FROM (
SELECT
table_name,
pg_table_size(table_name) AS table_size,
pg_indexes_size(table_name) AS indexes_size,
pg_total_relation_size(table_name) AS total_size
FROM (
SELECT ('"' || table_schema || '"."' || table_name || '"') AS table_name
FROM information_schema.tables where table_schema = 'public'
```

```

) AS all_tables
ORDER BY total_size DESC
) AS pretty_sizes
;

```

table_name	table_size	indexes_size	total_size
"public"."sbtest37"	21 GB	4938 MB	25 GB
"public"."sbtest29"	21 GB	4935 MB	25 GB
"public"."sbtest13"	21 GB	4935 MB	25 GB
"public"."sbtest71"	21 GB	4935 MB	25 GB
"public"."sbtest1"	21 GB	4935 MB	25 GB
"public"."sbtest54"	21 GB	4935 MB	25 GB
"public"."sbtest19"	21 GB	4935 MB	25 GB
"public"."sbtest63"	21 GB	4934 MB	25 GB
"public"."sbtest98"	21 GB	4934 MB	25 GB
"public"."sbtest16"	21 GB	4933 MB	25 GB
"public"."sbtest73"	21 GB	4933 MB	25 GB
"public"."sbtest92"	21 GB	4933 MB	25 GB
"public"."sbtest93"	21 GB	4933 MB	25 GB
"public"."sbtest55"	21 GB	4933 MB	25 GB
"public"."sbtest66"	21 GB	4933 MB	25 GB
"public"."sbtest68"	21 GB	4932 MB	25 GB
"public"."sbtest86"	21 GB	4932 MB	25 GB
"public"."sbtest31"	21 GB	4932 MB	25 GB
"public"."sbtest57"	21 GB	4932 MB	25 GB
"public"."sbtest39"	21 GB	4932 MB	25 GB
"public"."sbtest36"	21 GB	4932 MB	25 GB
"public"."sbtest12"	21 GB	4932 MB	25 GB
"public"."sbtest11"	21 GB	4931 MB	25 GB
"public"."sbtest8"	21 GB	4931 MB	25 GB
"public"."sbtest48"	21 GB	4931 MB	25 GB
"public"."sbtest5"	21 GB	4931 MB	25 GB
"public"."sbtest14"	21 GB	4931 MB	25 GB
"public"."sbtest20"	21 GB	4931 MB	25 GB
"public"."sbtest28"	21 GB	4931 MB	25 GB
"public"."sbtest18"	21 GB	4931 MB	25 GB
"public"."sbtest49"	21 GB	4931 MB	25 GB
"public"."sbtest99"	21 GB	4930 MB	25 GB
"public"."sbtest21"	21 GB	4930 MB	25 GB
"public"."sbtest95"	21 GB	4930 MB	25 GB
"public"."sbtest84"	21 GB	4930 MB	25 GB
"public"."sbtest47"	21 GB	4930 MB	25 GB
"public"."sbtest79"	21 GB	4930 MB	25 GB
"public"."sbtest42"	21 GB	4930 MB	25 GB
"public"."sbtest89"	21 GB	4930 MB	25 GB
"public"."sbtest33"	21 GB	4930 MB	25 GB
"public"."sbtest43"	21 GB	4930 MB	25 GB
"public"."sbtest67"	21 GB	4929 MB	25 GB
"public"."sbtest26"	21 GB	4929 MB	25 GB
"public"."sbtest88"	21 GB	4929 MB	25 GB
"public"."sbtest96"	21 GB	4929 MB	25 GB
"public"."sbtest94"	21 GB	4929 MB	25 GB
"public"."sbtest51"	21 GB	4929 MB	25 GB

- sysbench 测试数据整理

lua	threads	友商标盘(FF=75)				csd2000(FF=75)				qps ratio	rt ratio
		tps	ps(友商标盘)	rt(友商标盘)	rt(99th)	tps	qps(csd2000)	rt(csd2000)	rt(99th)		
oltp_read only	1	163	2.608	6.13	9.56	134	2.146	7.45	9.39	-17.7%	-17.7%
	4	539	8.626	7.42	16.71	495	7.912	8.09	10.46	-8.3%	-8.3%
	8	861	13.773	9.29	20.37	914	14.624	8.75	11.45	6.2%	6.2%
	16	1,236	19.774	12.94	28.67	1,441	23.062	11.1	19.65	16.6%	16.6%
	32	1,314	21.021	24.35	55.82	1,990	31.838	16.08	26.68	51.5%	51.4%
	64	1,409	22.546	45.41	86	2,283	36.524	28.02	41.1	62.0%	62.1%
	128	1,416	22.653	90.35	161.51	2,160	34.560	59.21	158.63	52.6%	52.6%
oltp_write only	256	1,514	24.226	168.9	1069.86	2,065	33.038	123.76	176.73	36.4%	36.5%
	1	620	3.723	1.61	5.00	552	3.313	1.81	2.91	-11.0%	-11.0%
	4	2,115	12.687	1.89	6.21	2,085	12.509	1.92	3.36	-1.4%	-1.6%
	8	3,006	18.035	2.66	9.39	3,698	22.186	2.16	4.25	23.0%	23.1%
	16	3,962	23.773	4.04	14.46	6,245	37.469	2.56	5.88	57.6%	57.8%
	32	4,719	28.315	6.78	18.95	6,402	38.412	5	28.67	35.7%	35.6%
	64	4,737	28.420	13.51	89.16	6,352	38.111	10.07	62.19	34.1%	34.2%
oltp_read write	128	4,462	26.772	28.67	292.6	6,175	37.049	20.72	121.08	38.4%	38.4%
	256	4,565	27.389	56.01	520.62	6,125	36.750	41.74	121.08	34.2%	34.2%
	1	123	2.464	8.12	16.41	107	2.144	9.33	12.30	-13.0%	-13.0%
	4	417	8.341	9.59	20.00	394	7.887	10.14	12.98	-5.4%	-5.4%
	8	712	14.242	11.23	23.1	722	14.434	11.08	14.46	1.3%	1.4%
	16	895	17.901	17.87	41.85	1,066	21.315	15.01	25.74	19.1%	19.1%
	32	1,050	21.006	30.46	63.32	1,593	31.856	20.08	32.53	51.6%	51.7%
	64	1,121	22.409	57.09	99.33	1,965	39.305	32.55	47.47	75.4%	75.4%
	128	1,120	22.401	114.16	186.54	1,756	35.117	72.83	277.21	56.8%	56.7%
	256	1,132	22.626	225.88	1352.03	1,639	32.780	155.79	219.36	44.9%	45.0%

3.3.3. BenchmarkSQL TPC-C 测试数据

■ CSD 2000 磁盘压缩比

观察 openGauss 数据库文件所在 CSD 磁盘的压缩比，3.12 左右。

```
[root@sfx111188 ~]# sh csd_size.sh sfdv1n1
Time - 2022-02-19-12:20:12
Device - /dev/sfdv1n1
Total capacity - 3577 GiB
Used space - 602 GiB
Free space - 2975 GiB
Logical data size - 1882 GiB
Compression ratio - 3.12 (logical data size / used space)
```

■ 表的大小

```
--查出表大小按大小排序并分离 data 与 index
SELECT
table_name,
pg_size_pretty(table_size) AS table_size,
pg_size_pretty(indexes_size) AS indexes_size,
pg_size_pretty(total_size) AS total_size
FROM (
SELECT
table_name,
pg_table_size(table_name) AS table_size,
```

```

pg_indexes_size(table_name) AS indexes_size,
pg_total_relation_size(table_name) AS total_size
FROM (
SELECT ('"' || table_schema || '"."' || table_name || '"') AS table_name
FROM information_schema.tables where table_schema = 'public'
) AS all_tables
ORDER BY total_size DESC
) AS pretty_sizes

```

table_name	table_size	indexes_size	total_size
"public"."bmsql_stock"	693 GB	21 GB	714 GB
"public"."bmsql_order_line"	600 GB	96 GB	696 GB
"public"."bmsql_customer"	367 GB	24 GB	391 GB
"public"."bmsql_oorder"	39 GB	19 GB	58 GB
"public"."bmsql_history"	53 GB	0 bytes	53 GB
"public"."bmsql_new_order"	7764 MB	2750 MB	10 GB
"public"."bmsql_district"	25 MB	2208 kB	27 MB
"public"."bmsql_item"	10 MB	2208 kB	12 MB
"public"."bmsql_warehouse"	2392 kB	240 kB	2632 kB
"public"."bmsql_config"	8192 bytes	16 kB	24 kB

(10 rows)

参考

CSD 2000 的产品详情、原理请参考下面资源获取。

- ScaleFlux 官网: [ScaleFlux Computational Storage - ScaleFlux Computational Storage](#)
- ScaleFlux 博客: [博客 - 博客 - ScaleFlux Computational Storage](#)
- ScaleFlux 公众号:

