LDBC

The graph & RDF
benchmark reference

Full Disclosure Report of the LDBC Social Network Benchmark

Audit of the LDBC Social Network Benchmark's
Business Intelligence Workload over TigerGraph

November 9, 2022

General Terms

Executive Summary

This report documents an audited execution of the LDBC SNB BI (Social Network Benchmark Business Intelligence) workload for TigerGraph.

TigerGraph is a massively parallel processing (MPP) graph database management system, designed for handling hybrid transaction/analytical processing (HTAP) query workloads. It is a distributed platform using a native graph storage format with an edge cut partitioning strategy. Within this, each segment (partition) of the graph holds a similar amount of vertices and processes requests in parallel. TigerGraph offers GSQL, a Turing-complete query language which provides both declarative features (e.g. graph patterns) as well as imperative ones (e.g. for expressing iterative graph algorithms with loops and accumulator primitives).

A note from the auditor on query languages

With the requirement on this benchmark of having all queries written in a domain-specific language, both the auditor and LDBC are aware there may be some comments on the imperative elements which are present within GSQL. As such, the auditor thought it sensible to provide his thoughts on this for the first BI audit:

> Upon initial review of the GSQL BI queries one can easily be drawn towards to the imperative elements of the language rooted in monoids (MapAccum, HeapAccum, SumAccum, etc.) which pull one away from any SQL-like functionality. This is especially the case if one's background is rooted in the world of SQL/Cypher. However, as I progressed through my review, these elements begin to feel like a natural way of defining the complex steps of each query and did not feel like something a user of the platform would struggle to learn, with some time and training.
>
> Instead of focusing on one language in particular, I believe this audit raises many design questions for query languages in general for how to represent complex graph functionality (such as those raised within the benchmark) in a truly declarative fashion.

Declaration of Audit Success

This report contains details of a successful execution of the LDBC SNB BI benchmark. The results have been gathered by an independent auditor who has validated the implementation of the queries and verified the system's configuration conforms to the description of the benchmark and its strict requirements.

Sponsorship and Funding Disclaimer

TigerGraph, as an LDBC member, are the Test Sponsor of this audit. The audit itself was funded by AMD who also provided the hardware upon which the audit was conducted. This arrangement was deemed acceptable by both parties, the LDBC Steering Committee and the Auditor.

DocuSigned by:
Andrei Fabian Murariu
435313E00FE040E...

11/9/2022

Andrei Fabian Murariu
(Auditor)

Date

Pometry
(Audit Company)

DocuSigned by:
Gabor Szarnyas
FA33C2C8CB944A5...

11/9/2022

Dr. Gabor Szarnyas
(Head of LDBC SNB Task Force)

Date

DocuSigned by:
Mingxi Wu
D302A312743E43D...

11/9/2022

Mingxi Wu
(Test Sponsor Representative)

Date

Table of Contents

1 Benchmark Description

The audit was conducted in compliance with the Social Network Benchmark's specification.

Table 1.1: Benchmark Overview

Artifact	Version	URL
Specification	2.2.0	`https://arxiv.org/pdf/2001.02299v7.pdf`
Data generator	0.5.0	`https://github.com/ldbc/ldbc_snb_datagen_spark/releases/tag/v0.5.0`
Driver and implementations	1.0.2	`https://github.com/ldbc/ldbc_snb_bi/releases/tag/v1.0.2`

2 System Description and Pricing Summary

2.1 Details of machines driving and running the workload

2.1.1 Machine overview

The hardware used for the experiments in this report was an unreleased Dell PowerEdge R6625 server provided by AMD. The benchmark framework and the TigerGraph DBMS were running on bare-metal, no virtualization of any kind was used.

Table 2.1: Machine Type

Hardware provider	AMD
Common name of the item	Dell PowerEdge R6625
Operating system	Ubuntu 20.04.5 LTS

2.1.2 CPU details

The details below were obtained using the commands `cat /proc/cpuinfo` (Listing A.1) and `lscpu` (Listing A.2).

Table 2.2: CPU details summary

Type	AMD® EPYC® 9354 32-Core Processor
Total number	2
Cores per CPU	32
Threads per CPU	64
Total threads	128
CPU clock frequency	3.748 GHz
Total cache size per CPU	L1d cache: 2MB L1i cache: 2MB L2 cache: 64MB L3 cache: 512MB

2.1.3 Memory details

The total size of the memory installed is 1.5TB. This information was obtained using the `cat /proc/meminfo` (Listing A.3) and `lshw -C memory` (Listing A.4) commands.

2.1.4 Disk and storage details

Table 2.3: Disk details summary

Root disk	1 × Dell Ent NVMe v2 AGN RI U.2 1.92TB
Data disk	1 × Dell Ent NVMe v2 AGN RI U.2 3.84TB
Disk controller	Samsung Electronics Co Ltd NVMe SSD Controller PM173X
Root file system	ext4
Data file system	xfs

The 4KB QD1 write performance on the data disk was measured with the `fio` command and the output (Listing B.1) showed an average of 13 049.63 IOPS.

2.1.5 Network details

The presented benchmark run only used a single machine, thus network details are not included here.

2.1.6 Machine pricing

The system pricing summary in US dollars ($) is included in the table below. The full breakdown of the costings for the Dell PowerEdge R6625, provided by AMD, can be found in Appendix E.

Table 2.4: Pricing summary

Item	**Price**
Dell PowerEdge R6625	$146,307.09
Software license (3 years)	$1,097,280.00
Maintenance fee (3-year contract)	$109,728.00
Total cost	$1,353,315.09

2.1.7 System version and availability

Table 2.5: System versions

System	**Version**	**License**
TigerGraph	3.7.0	Enterprise Licence provided by TigerGraph

3 Dataset Generation

3.1 General information

The data generation settings of the LDBC Datagen are described below.

Table 3.1: Datagen settings summary

Data format for TigerGraph	composite-projected-fk layout, compressed CSV files
Scale factors for TigerGraph	10 (validation), 1000 (benchmark)
Data format for Neo4j	composite-projected-fk layout, compressed CSV files with quoted fields and without headers
Scale factors for Neo4j	10 (validation)

3.2 Datagen configurations

The datasets and query substitution parameters used for the benchmark and the cross-validation runs were retrieved from the following URLs. The URLs are served by LDBC's official data repository, available as a public bucket in the Cloudflare R2 object storage.[1]

3.2.1 SF10

- `https://pub-383410a98aef4cb686f0c7601eddd25f.r2.dev/bi-pre-audit/parameters-2022-10-01.zip`
- `https://pub-383410a98aef4cb686f0c7601eddd25f.r2.dev/bi-pre-audit/bi-sf10-composite-projected-fk-with-quotes-without-headers.tar.zst`
- `https://pub-383410a98aef4cb686f0c7601eddd25f.r2.dev/bi-pre-audit/bi-sf10-composite-projected-fk.tar.zst`

3.2.2 SF1000

- `https://pub-383410a98aef4cb686f0c7601eddd25f.r2.dev/bi-pre-audit/parameters-2022-10-01.zip`
- `https://pub-383410a98aef4cb686f0c7601eddd25f.r2.dev/bi-pre-audit/bi-sf1000-composite-projected-fk.tar.zst.000`
- `https://pub-383410a98aef4cb686f0c7601eddd25f.r2.dev/bi-pre-audit/bi-sf1000-composite-projected-fk.tar.zst.001`
- `https://pub-383410a98aef4cb686f0c7601eddd25f.r2.dev/bi-pre-audit/bi-sf1000-composite-projected-fk.tar.zst.002`

To re-generate these datasets from scratch, use the instructions provided in Appendix C.

3.3 Data loading and data schema

The data preprocessing and loading times are reported below. Values were measured using the GNU Time tool (`/usr/bin/time`) with the `-v` flag, reading the *Elapsed (wall clock) time* from the output. The column **Data preprocessing time** shows how much time it took to preprocess the CSV files. For this benchmark execution, the preprocessing only consisted of decompressing the `.csv.gz` files. The column **Data loading time** shows how long it took to create a graph from the input CSV files and perform the initial indexing. The column **Total time** contains the sum of the data preprocessing and loading times.

The TigerGraph data schema is shown in Listing D.1.

Table 3.2: Data preprocessing and loading times for TigerGraph on scale factor 1000

Scale factor	Data preprocessing time (s)	Data loading time (s)	Total time (s)
1000	268.00	697.42	965.42

[1] `https://www.cloudflare.com/products/r2/`

4 Implementation Details

4.1 Execution mode

Section 7.5.2.2 of the SNB specification defines two execution modes for the *throughput batches*. In *disjoint read-write mode*, the updates for each day of the benchmark's simulation are applied in bulk, separately from the read queries (i.e. there are no overlapping read and write operations). In *concurrent read-write mode*, the updates are applied concurrently with the reads. Systems opting for concurrent read-write mode are subject to the LDBC ACID test[1].

In the current audited run, TigerGraph was executed using the *disjoint read-write mode*. Therefore, no ACID tests were conducted.

4.2 Use of auxiliary data structures

The TigerGraph implementation precomputes the following auxiliary data structures. These are executed in each batch after the writes have been applied.

- **Root Post:** For each Message node (Comments and Posts), an edge to the corresponding Message thread's root Post is inserted. These derived edges are maintained incrementally, i.e. root Post edges are inserted for newly inserted Messages and removed for deleted Messages.
- **Q4:** For each Forum, the maximum number of members (for number of members per country) is precomputed.
- **Q6:** For each Message, the popularityScore defined in the query is precomputed.
- **Q14:** The weight attributed on the knows edges are precomputed based on the number of interactions between the two Person nodes.
- **Q19:** The weight attributes on the knows edges are precomputed based on the number of interactions between the two Person nodes.
- **Q20:** The weight attributes on the knows edges are precomputed based on the classYear attributes on the studyAt edges that point to the same University from the endpoint Person nodes.

The precomputations for Q14 and Q19 are performed together using different scoring methods for establishing the edge weights. We display the runtime of this operation as "precomputation for Q14 and Q19" in Table 5.3.

4.3 Benchmark execution

The benchmark is executed using the following commands.
Note: despite what the script's name suggests, this is benchmark was executed bare-metal with the TigerGraph instance running on the server machine and not using any containerization/virtualization technology.

```
cd /data/ldbc_snb_bi/tigergraph
# change the following lines in k8s/vars.sh
# export NUM_NODES=1 # number of pods or nodes
# export SF=1000
# export TG_DATA_DIR=/data/sf${SF}
# export TG_PARAMETER=/data/parameters-sf${SF}

nohup ./k8s/setup.sh > log.setup 2>&1 < /dev/null &
tail -f log.setup

nohup ./k8s/benchmark.sh > log.benchmark 2>&1 < /dev/null &
```

[1] https://github.com/ldbc/ldbc_acid

```
tail -f log.setup
```

Listing 4.1: Script to execute the benchmark on TigerGraph for SF1000

5 Performance Results

5.1 TigerGraph performance results

Table 5.1: Summary of results for TigerGraph on scale factor 1000

Benchmark duration	**Power@SF**	**Power@SF/$**	**Throughput@SF**	**Throughput@SF/$**
198.38 minutes	30 990.08	22.90	12 993.85	9.60

Table 5.2: Detailed **power test results** for TigerGraph on scale factor 1000. Execution times are reported in seconds.

Query	**Count**	**Min.**	**Max.**	**Mean**	$\mathbf{P_{50}}$	$\mathbf{P_{90}}$	$\mathbf{P_{95}}$	$\mathbf{P_{99}}$
1	30	8.808	16.038	9.476	9.194	9.574	9.612	9.623
2a	30	1.747	18.506	6.990	4.316	15.922	16.169	16.493
2b	30	0.596	3.248	1.735	1.500	2.818	2.946	3.102
3	30	2.244	9.775	4.148	3.229	7.653	7.961	9.408
4	30	1.464	1.593	1.503	1.503	1.523	1.527	1.529
5	30	1.083	1.315	1.127	1.119	1.142	1.145	1.145
6	30	0.900	1.020	0.955	0.957	0.983	0.990	0.994
7	30	1.924	2.318	2.104	2.107	2.200	2.218	2.296
8a	30	1.671	2.611	1.939	1.874	2.168	2.177	2.589
8b	30	0.836	1.068	0.872	0.862	0.917	0.926	0.934
9	30	5.865	7.709	6.265	6.122	6.823	7.095	7.534
10a	30	4.052	9.476	6.128	6.199	7.188	7.221	7.684
10b	30	2.230	3.032	2.716	2.788	3.007	3.023	3.027
11	30	3.182	3.856	3.513	3.503	3.690	3.763	3.856
12	30	5.562	9.919	7.433	8.352	9.170	9.324	9.440
13	30	11.674	12.167	11.863	11.820	12.061	12.071	12.164
14a	30	7.432	8.292	7.972	8.001	8.089	8.115	8.221
14b	30	2.859	2.985	2.900	2.899	2.918	2.932	2.962
15a	30	12.607	15.661	13.452	13.376	13.961	13.996	14.178
15b	30	10.587	43.968	29.038	26.693	38.828	40.241	43.533
16a	30	4.281	12.465	6.649	6.555	8.453	8.788	9.834
16b	30	1.448	3.407	1.930	1.578	3.004	3.214	3.215
17	30	2.463	2.794	2.687	2.717	2.762	2.764	2.764
18	30	10.955	14.149	11.829	11.693	12.502	12.525	12.989
19a	30	2.176	3.204	2.862	3.004	3.112	3.137	3.182
19b	30	2.229	3.769	3.041	3.066	3.260	3.302	3.416
20a	30	0.416	2.473	1.046	0.420	2.271	2.275	2.470
20b	30	0.617	1.432	0.884	0.823	1.231	1.231	1.231

Table 5.3: Operations in the **power test** for TigerGraph on scale factor 1000. Execution times are reported in seconds. Root Post precomputations are performed for each Comment insertion and deletion operation, therefore, they are reported as part of the writes.

Operation	**Time**
total read time	4 591.903
total write time	1 283.052
precomputation for Q4	66.929
precomputation for Q6	68.981
precomputation for Q14 and Q19	596.686
precomputation for Q20	22.984

6 Validation of the Results

The results were cross-validated against the Neo4j reference implementation[1] on scale factor 10, using Neo4j Community Edition, version 4.4.2. Neo4j is a graph database management system where data is stored using the property graph data model and is queried using the Cypher language. Note, whilst the validation queries for this audit were implemented predominantly in Cypher, some path queries make use of the official Neo4j "Graph Data Science" library plugin.

Listing 6.1: Output of the Neo4j–TigerGraph cross-validation command

```
$ export SF=10
$ scripts/cross-validate.sh cypher tigergraph

+++  Files "cypher/output/output-sf10/results.csv" and "tigergraph/output/output-sf10/results.csv" are equal
```

[1] https://github.com/ldbc/ldbc_snb_bi/tree/21e3cf176ea0ab1fb1cde7691845711fe23f2fe4/cypher

7 Supplementary Materials

Table 7.1: Supplementary materials

File or Directory	Purpose
`cypher/output/output-sf10`	Output of the Neo4j reference implementation
`parameters/parameters-sf{10,1000}.tar`	Query substitution parameters
`scoring/calculate-scores.py`	Python script to calculate the scores of the benchmark run
`tigergraph/output/output-sf{10,1000}`	Benchmark logs and outputs
`ldbc_snb_bi-1.0.2.tar.gz`	Benchmark driver and reference implementations
`ldbc_snb_datagen_spark-0.5.1.tar.gz`	Data generator
`ldbc_snb_specification-2.2.0.pdf`	Benchmark specification

The `ldbc_snb_bi_tigergraph_sf1000_attachments` folder's directory structure is as follows:

```
ldbc_snb_bi_tigergraph_sf1000_attachments
├── cypher
│   └── output
│       └── output-sf10
│           └── results.csv
├── parameters
│   ├── parameters-sf10.tar
│   └── parameters-sf1000.tar
├── scoring
│   └── calculate-scores.py
├── tigergraph
│   └── output
│       ├── output-sf10
│       │   ├── benchmark.csv
│       │   ├── load.csv
│       │   ├── log.benchmark
│       │   ├── log.setup
│       │   ├── log.validate_benchmark
│       │   ├── results.csv
│       │   └── timings.csv
│       └── output-sf1000
│           ├── benchmark.csv
│           ├── load.csv
│           ├── log.benchmark
│           ├── log.setup
│           ├── results.csv
│           └── timings.csv
├── ldbc_snb_bi-1.0.2.tar.gz
├── ldbc_snb_datagen_spark-0.5.1.tar.gz
└── ldbc_snb_specification-2.2.0.pdf
```

A CPU and Memory details

Listing A.1: Output of the `cat /proc/cpuinfo` command for a single CPU core

```
processor       : 0
vendor_id       : AuthenticAMD
cpu family      : 25
model           : 17
model name      : AMD EPYC 9354 32-Core Processor
stepping        : 1
cpu MHz         : 3748.221
cache size      : 1024 KB
physical id     : 0
siblings        : 64
core id         : 0
cpu cores       : 32
apicid          : 0
initial apicid  : 0
fpu             : yes
fpu_exception   : yes
cpuid level     : 16
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ht syscall nx mmxext fxsr_opt pdpe1gb rdtscp lm constant
_tsc rep_good nopl nonstop_tsc cpuid extd_apicid aperfmperf pni pclmulqdq monitor ssse3 fma cx16 pcid sse4_1 sse4_2 x2apic movbe popcnt aes xsave avx f16c rdrand lahf_lm cm
p_legacy svm extapic cr8_legacy abm sse4a misalignsse 3dnowprefetch osvw ibs skinit wdt tce topoext perfctr_core perfctr_nb bpext perfctr_llc mwaitx cpb cat_l3 cdp_l3 invpc
id_single hw_pstate ssbd mba ibrs ibpb stibp vmmcall fsgsbase bmi1 avx2 smep bmi2 erms invpcid cqm rdt_a avx512f avx512dq rdseed adx smap avx512ifma clflushopt clwb avx512c
d sha_ni avx512bw avx512vl xsaveopt xsavec xgetbv1 xsaves cqm_llc cqm_occup_llc cqm_mbm_total cqm_mbm_local avx512_bf16 clzero irperf xsaveerptr wbnoinvd arat npt lbrv svm_
lock nrip_save tsc_scale vmcb_clean flushbyasid decodeassists pausefilter pfthreshold avic v_vmsave_vmload vgif avx512vbmi umip pku ospke avx512_vbmi2 gfni vaes vpclmulqdq
avx512_vnni avx512_bitalg avx512_vpopcntdq rdpid overflow_recov succor smca flush_l1d
bugs            : sysret_ss_attrs spectre_v1 spectre_v2 spec_store_bypass
bogomips        : 6502.05
TLB size        : 3584 4K pages
clflush size    : 64
cache_alignment : 64
address sizes   : 52 bits physical, 57 bits virtual
power management: ts ttp tm hwpstate cpb eff_freq_ro [13] [14]
```

Listing A.2: Output of the `lscpu` command

```
Architecture:                    x86_64
CPU op-mode(s):                  32-bit, 64-bit
Byte Order:                      Little Endian
Address sizes:                   52 bits physical, 57 bits virtual
CPU(s):                          128
On-line CPU(s) list:             0-127
Thread(s) per core:              2
Core(s) per socket:              32
Socket(s):                       2
NUMA node(s):                    2
Vendor ID:                       AuthenticAMD
CPU family:                      25
Model:                           17
```

```
Model name:                      AMD EPYC 9354 32-Core Processor
Stepping:                        1
Frequency boost:                 enabled
CPU MHz:                         3784.277
CPU max MHz:                     3250.0000
CPU min MHz:                     1500.0000
BogoMIPS:                        6502.05
Virtualization:                  AMD-V
L1d cache:                       2 MiB
L1i cache:                       2 MiB
L2 cache:                        64 MiB
L3 cache:                        512 MiB
NUMA node0 CPU(s):               0-31,64-95
NUMA node1 CPU(s):               32-63,96-127
Vulnerability Itlb multihit:     Not affected
Vulnerability L1tf:              Not affected
Vulnerability Mds:               Not affected
Vulnerability Meltdown:          Not affected
Vulnerability Mmio stale data:   Not affected
Vulnerability Spec store bypass: Mitigation; Speculative Store Bypass disabled via prctl and seccomp
Vulnerability Spectre v1:        Mitigation; usercopy/swapgs barriers and __user pointer sanitization
Vulnerability Spectre v2:        Mitigation; Retpolines, IBPB conditional, IBRS_FW, STIBP always-on, RSB filling,
      PBRSB-eIBRS Not affected
Vulnerability Srbds:             Not affected
Vulnerability Tsx async abort:   Not affected
Flags:                           fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush
     mmx fxsr sse sse2 ht syscall nx mmxext fxsr_opt pdpe1gb rdtscp lm constant_tsc rep_good nopl nonstop_tsc
     cpuid extd_apicid aperfmperf pni pclmulqdq monitor ssse3 fma cx16 pcid sse4_1 sse4_2 x2apic movbe popcnt aes
     xsave avx f16c rdrand lahf_lm cmp_legacy svm extapic cr8_legacy abm sse4a misalignsse 3dnowprefetch osvw ibs
     skinit wdt tce topoext perfctr_core perfctr_nb bpext perfctr_llc mwaitx cpb cat_l3 cdp_l3 invpcid_single
     hw_pstate ssbd mba ibrs ibpb stibp vmmcall fsgsbase bmi1 avx2 smep bmi2 erms invpcid cqm rdt_a avx512f
     avx512dq rdseed adx smap avx512ifma clflushopt clwb avx512cd sha_ni avx512bw avx512vl xsaveopt xsavec xgetbv1
      xsaves cqm_llc cqm_occup_llc cqm_mbm_total cqm_mbm_local avx512_bf16 clzero irperf xsaveerptr wbnoinvd arat
     npt lbrv svm_lock nrip_save tsc_scale vmcb_clean flushbyasid decodeassists pausefilter pfthreshold avic
     v_vmsave_vmload vgif avx512vbmi umip pku ospke avx512_vbmi2 gfni vaes vpclmulqdq avx512_vnni avx512_bitalg
     avx512_vpopcntdq rdpid overflow_recov succor smca flush_l1d
```

Listing A.3: Output of the `cat /proc/meminfo` command

```
MemTotal:       1584941904 kB
MemFree:        13297692 kB
MemAvailable:   1553362564 kB
Buffers:          587100 kB
Cached:         1540386604 kB
SwapCached:           20 kB
Active:         190172740 kB
Inactive:       1370199800 kB
Active(anon):   17436616 kB
Inactive(anon):  2371644 kB
Active(file):   172736124 kB
Inactive(file): 1367828156 kB
Unevictable:      497352 kB
Mlocked:          497352 kB
SwapTotal:       8388604 kB
SwapFree:        8369916 kB
Dirty:               480 kB
Writeback:             0 kB
AnonPages:      19851976 kB
```

```
Mapped:           389400 kB
Shmem:            419148 kB
KReclaimable:    7653120 kB
Slab:            9298372 kB
SReclaimable:    7653120 kB
SUnreclaim:      1645252 kB
KernelStack:       66864 kB
PageTables:        94028 kB
NFS_Unstable:          0 kB
Bounce:                0 kB
WritebackTmp:          0 kB
CommitLimit:    800859556 kB
Committed_AS:   29759712 kB
VmallocTotal:   34359738367 kB
VmallocUsed:      509280 kB
VmallocChunk:          0 kB
Percpu:           228864 kB
HardwareCorrupted:     0 kB
AnonHugePages:         0 kB
ShmemHugePages:        0 kB
ShmemPmdMapped:        0 kB
FileHugePages:         0 kB
FilePmdMapped:         0 kB
CmaTotal:              0 kB
CmaFree:               0 kB
HugePages_Total:       0
HugePages_Free:        0
HugePages_Rsvd:        0
HugePages_Surp:        0
Hugepagesize:       2048 kB
Hugetlb:               0 kB
DirectMap4k:     1931836 kB
DirectMap2M:    313243648 kB
DirectMap1G:    1294991360 kB
```

Listing A.4: Output of the `lshw -C memory` command

```
# there are 24 memory banks populated with 64GB modules
     *-bank:23
          description: DIMM Synchronous Registered (Buffered) 4800 MHz (0.2 ns)
          product: M321R8GA0BB0-CQKEG
          vendor: Samsung
          physical id: 17
          serial: 02F9CFC4
          slot: B12
          size: 64GiB
          width: 64 bits
          clock: 505MHz (2.0ns)
```

B IO PERFORMANCE

Listing B.1: Output of the `fio` command

```
iotest: (g=0): rw=write, bs=(R) 4096B-4096B, (W) 4096B-4096B, (T) 4096B-4096B, ioengine=sync, iodepth=1
fio-3.16
Starting 1 process
iotest: Laying out IO file (1 file / 2048MiB)

iotest: (groupid=0, jobs=1): err= 0: pid=1746724: Tue Nov  1 13:58:58 2022
  write: IOPS=13.1k, BW=50.0MiB/s (53.5MB/s)(2048MiB/40171msec); 0 zone resets
    clat (usec): min=16, max=2167, avg=20.47, stdev=12.00
     lat (usec): min=16, max=2167, avg=20.51, stdev=12.00
    clat percentiles (usec):
     |  1.00th=[   18],  5.00th=[   18], 10.00th=[   18], 20.00th=[   19],
     | 30.00th=[   19], 40.00th=[   20], 50.00th=[   20], 60.00th=[   20],
     | 70.00th=[   20], 80.00th=[   21], 90.00th=[   23], 95.00th=[   26],
     | 99.00th=[   44], 99.50th=[   63], 99.90th=[  155], 99.95th=[  208],
     | 99.99th=[  371]
   bw (  KiB/s): min=46528, max=59944, per=99.99%, avg=52198.55, stdev=2918.55, samples=80
   iops        : min=11632, max=14986, avg=13049.63, stdev=729.64, samples=80
  lat (usec)   : 20=71.39%, 50=27.83%, 100=0.54%, 250=0.21%, 500=0.03%
  lat (usec)   : 750=0.01%, 1000=0.01%
  lat (msec)   : 2=0.01%, 4=0.01%
  fsync/fdatasync/sync_file_range:
    sync (usec): min=38, max=27960, avg=55.76, stdev=142.16
    sync percentiles (usec):
     |  1.00th=[   41],  5.00th=[   42], 10.00th=[   42], 20.00th=[   43],
     | 30.00th=[   44], 40.00th=[   45], 50.00th=[   46], 60.00th=[   48],
     | 70.00th=[   49], 80.00th=[   51], 90.00th=[   57], 95.00th=[   71],
     | 99.00th=[  202], 99.50th=[  359], 99.90th=[ 1598], 99.95th=[ 2409],
     | 99.99th=[ 4948]
  cpu          : usr=1.06%, sys=15.73%, ctx=1048866, majf=0, minf=23
  IO depths    : 1=200.0%, 2=0.0%, 4=0.0%, 8=0.0%, 16=0.0%, 32=0.0%, >=64=0.0%
     submit    : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
     complete  : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
     issued rwts: total=0,524288,0,0 short=524287,0,0,0 dropped=0,0,0,0
     latency   : target=0, window=0, percentile=100.00%, depth=1

Run status group 0 (all jobs):
  WRITE: bw=50.0MiB/s (53.5MB/s), 50.0MiB/s-50.0MiB/s (53.5MB/s-53.5MB/s), io=2048MiB (2147MB), run=40171-40171msec

Disk stats (read/write):
    dm-2: ios=0/2622676, merge=0/0, ticks=0/50156, in_queue=50156, util=99.79%, aggrios=0/0, aggrmerge=0/0, aggrticks=0/0, aggrin_queue=0, aggrutil=0.00%
  nvme3n1: ios=0/0, merge=0/0, ticks=0/0, in_queue=0, util=0.00%
```

C Dataset generation instructions

The datasets can be generated using the LDBC SNB Datagen. To regenerate the data sets used in this benchmark, build the Datagen JAR as described in the project's README, configure the AWS EMR environment, upload the JAR to the S3 bucket (denoted as `${BUCKET_NAME}`) and run the following commands to generate the datasets used in this audit.

Note that while the datasets for TigerGraph were generated as gzip-compressed archives, they are decompressed during preprocessing. Decompressing the SF1000 data set took 4 minutes 28 seconds (wall clock) when performed by the following command: `time find /data/sf1000 -name "*.csv.gz" -print0 | parallel -q0 gunzip`

Listing C.1: Script to generate the SF10 dataset for TigerGraph in AWS EMR. This dataset is only used for cross-validation

```
export SCALE_FACTOR=10
export JOB_NAME=sf${SCALE_FACTOR}-projected-csv-gz

./tools/emr/submit_datagen_job.py \
    --use-spot \
    --bucket ${BUCKET_NAME} \
    --copy-all \
    --az us-east-2c \
    ${JOB_NAME} \
    ${SCALE_FACTOR} \
    csv \
    bi \
    -- \
    --explode-edges \
    --format-options compression=gzip \
    --generate-factors
```

Listing C.2: Script to generate the SF1000 dataset for TigerGraph in AWS EMR. This dataset is used for the benchmark run

```
export SCALE_FACTOR=1000
export JOB_NAME=sf${SCALE_FACTOR}-projected-csv-gz

./tools/emr/submit_datagen_job.py \
    --use-spot \
    --bucket ${BUCKET_NAME} \
    --copy-all \
    --az us-east-2c \
    ${JOB_NAME} \
    ${SCALE_FACTOR} \
    csv \
    bi \
    -- \
    --explode-edges \
    --format-options compression=gzip \
    --generate-factors
```

Listing C.3: Script to generate the SF10 dataset for Neo4j locally. This dataset is only used for cross-validation

```
export SCALE_FACTOR=10
export LDBC_SNB_DATAGEN_MAX_MEM=60G
export LDBC_SNB_DATAGEN_JAR=$(sbt -batch -error 'print assembly / assemblyOutputPath')

tools/run.py \
    --cores $(nproc) \
    --memory ${LDBC_SNB_DATAGEN_MAX_MEM} \
    -- \
    --format csv \
    --scale-factor ${SCALE_FACTOR} \
    --explode-edges \
    --mode bi \
    --output-dir out-sf${SCALE_FACTOR}/ \
    --generate-factors \
    --format-options header=false,quoteAll=true,compression=gzip
```

D Data schema

Listing D.1: Content of the GSQL schema used by TigerGraph

```
## Message
CREATE VERTEX Comment (PRIMARY_ID id UINT, creationDate INT, locationIP STRING, browserUsed STRING, content STRING, length UINT) WITH primary_id_as_attribute="TRUE"
CREATE VERTEX Post (PRIMARY_ID id UINT, imageFile STRING, creationDate INT, locationIP STRING, browserUsed STRING, language STRING, content STRING, length UINT) WITH primary_id_as_attribute="TRUE"
## organisation
CREATE VERTEX Company (PRIMARY_ID id UINT, name STRING, url STRING) WITH primary_id_as_attribute="TRUE"
CREATE VERTEX University (PRIMARY_ID id UINT, name STRING, url STRING) WITH primary_id_as_attribute="TRUE"
## place
CREATE VERTEX City (PRIMARY_ID id UINT, name STRING, url STRING) WITH primary_id_as_attribute="TRUE"
CREATE VERTEX Country (PRIMARY_ID id UINT, name STRING, url STRING) WITH primary_id_as_attribute="TRUE"
CREATE VERTEX Continent (PRIMARY_ID id UINT, name STRING, url STRING) WITH primary_id_as_attribute="TRUE"
## etc
CREATE VERTEX Forum (PRIMARY_ID id UINT, title STRING, creationDate INT,
    maxMember UINT) WITH primary_id_as_attribute="TRUE" // maxMember is for precompute in BI-4
CREATE VERTEX Person (PRIMARY_ID id UINT, firstName STRING, lastName STRING, gender STRING, birthday INT, creationDate INT, locationIP STRING, browserUsed STRING, speaks SET<STRING>, email SET<STRING>,
    popularityScore UINT) WITH primary_id_as_attribute="TRUE" // popularityScore is for precompute in BI-6
CREATE VERTEX Tag (PRIMARY_ID id UINT, name STRING, url STRING) WITH primary_id_as_attribute="TRUE"
CREATE VERTEX TagClass (PRIMARY_ID id UINT, name STRING, url STRING) WITH primary_id_as_attribute="TRUE"


# create edge
CREATE DIRECTED EDGE CONTAINER_OF (FROM Forum, TO Post) WITH REVERSE_EDGE="CONTAINER_OF_REVERSE"
CREATE DIRECTED EDGE HAS_CREATOR (FROM Comment|Post, TO Person) WITH REVERSE_EDGE="HAS_CREATOR_REVERSE"
CREATE DIRECTED EDGE HAS_INTEREST (FROM Person, TO Tag) WITH REVERSE_EDGE="HAS_INTEREST_REVERSE"
CREATE DIRECTED EDGE HAS_MEMBER (FROM Forum, TO Person, creationDate INT) WITH REVERSE_EDGE="HAS_MEMBER_REVERSE"
CREATE DIRECTED EDGE HAS_MODERATOR (FROM Forum, TO Person) WITH REVERSE_EDGE="HAS_MODERATOR_REVERSE"
CREATE DIRECTED EDGE HAS_TAG (FROM Comment|Post|Forum, TO Tag) WITH REVERSE_EDGE="HAS_TAG_REVERSE"
CREATE DIRECTED EDGE HAS_TYPE (FROM Tag, TO TagClass) WITH REVERSE_EDGE="HAS_TYPE_REVERSE"
CREATE DIRECTED EDGE IS_LOCATED_IN (FROM Company, TO Country | FROM Person, TO City | FROM University, TO City) WITH REVERSE_EDGE="IS_LOCATED_IN_REVERSE"
CREATE DIRECTED EDGE MESG_LOCATED_IN (FROM Comment, TO Country | FROM Post, TO Country) // Reverse edge of Comment/Post -IS_Located_IN-> Country will cause Country connected by too many edges, which makes loading slow
CREATE DIRECTED EDGE IS_PART_OF (FROM City, TO Country | FROM Country, TO Continent) WITH REVERSE_EDGE="IS_PART_OF_REVERSE"
CREATE DIRECTED EDGE IS_SUBCLASS_OF (FROM TagClass, TO TagClass) WITH REVERSE_EDGE="IS_SUBCLASS_OF_REVERSE"
CREATE UNDIRECTED EDGE KNOWS (FROM Person, TO Person, creationDate INT, weight19 UINT, weight20 UINT DEFAULT 10000)
CREATE DIRECTED EDGE LIKES (FROM Person, TO Comment|Post, creationDate INT) WITH REVERSE_EDGE="LIKES_REVERSE"
CREATE DIRECTED EDGE REPLY_OF (FROM Comment, TO Comment|Post) WITH REVERSE_EDGE="REPLY_OF_REVERSE"
CREATE DIRECTED EDGE STUDY_AT (FROM Person, TO University, classYear INT) WITH REVERSE_EDGE="STUDY_AT_REVERSE"
CREATE DIRECTED EDGE WORK_AT (FROM Person, TO Company, workFrom INT) WITH REVERSE_EDGE="WORK_AT_REVERSE"

CREATE DIRECTED EDGE ROOT_POST (FROM Comment, TO Post) WITH REVERSE_EDGE="ROOT_POST_REVERSE" //FOR BI-3,9,17
CREATE DIRECTED EDGE REPLY_COUNT (FROM Person, TO Person, cnt UINT)

CREATE GLOBAL SCHEMA_CHANGE JOB addIndex {
  ALTER VERTEX Country ADD INDEX country_name ON (name);
  ALTER VERTEX Company ADD INDEX company_name ON (name);
  ALTER VERTEX University ADD INDEX university_name ON (name);
  ALTER VERTEX Tag ADD INDEX tag_name ON (name);
  ALTER VERTEX TagClass ADD INDEX tagclass_name ON (name);
}

RUN GLOBAL SCHEMA_CHANGE JOB addIndex
CREATE GRAPH ldbc_snb (*)
```

E Machine Cost Breakdown

Quantity	Description	Cost
1	PowerEdge R6625 Server	$144,718.88
1	2.5 Chassis	$0.00
1	NVMe Backplane	$0.00
1	Trusted Platform Module 2.0 V3	$0.00
1	C03-03 : 8x U.2 G4 RAID - Low Z (FPERC 12)	$0.00
1	MOD, PRC, 9354, 2.7, GOA, 32C, XXX, QB	$0.00
1	MOD, PRC, 9354, 2.7, GOA, 32C, XXX, QB	$0.00
1	Performance Optimized	$0.00
1	4800MT/s RDIMMs	$0.00
24	32GB RDIMM, 4800MT/s Dual Rank	$0.00
1	C31, No RAID with NVMe and front PERC	$0.00
1	No Hard Drive	$0.00
1	1.92TB Enterprise NVMe Read Intensive AG Drive U.2 Gen4 with carrier	$0.00
7	3.84TB Enterprise NVMe Read Intensive AG Drive U.2 Gen4 with carrier	$0.00
1	Performance BIOS Settings	$0.00
1	UEFI BIOS Boot Mode with GPT Partition	$0.00
1	High Performance Fan for CPU greater than or equal to 180W(2 CPU)	$0.00
1	Dual, Hot-Plug, Power Supply 800W Redundant (1+1), L, Mixed Mode	$0.00
2	Power Cord - C13, 3M, 125V, 15A	$0.00
1	Riser Config 2, 1 x 16 LP PCIe slot (CPU1), 2 x 16 LP PCIeslot (CPU2)	$0.00
1	PowerEdge R6625 Motherboard	$0.00
1	Broadcom 5720 Dual Port 1GbE Optional LOM	$0.00
1	iDRAC9, Enterprise 15G	$0.00
1	Dell EMC Luggage Tag (x8 or x10 chassis)	$0.00
1	Standard Bezel	$0.00
1	No Quick Sync	$0.00
1	iDRAC, Factory Generated Password	$0.00
1	Ubuntu Support by Dell, 1yr Premium Sub, 1 Physical with Unlimited VMs	$0.00
1	ReadyRails Sliding Rails Without Cable Management Arm or Strain Relief Bar	$0.00
1	Cable Management Arm	$0.00
1	BOSS-S2 controller card + with 2 M.2 240GB (RAID 1)	$0.00
1	No Systems Documentation, NoOpenManage DVD Kit	$0.00
1	PowerEdge R6625 Shipping Material 4	$0.00
1	PowerEdge R6625 CCC Marking, No CE Marking	$0.00
1	US Order	$0.00
1	Basic Next Business Day 36 Months	$200.01
1	ProSupport and Next BusinessDay Onsite Service Initial	$1,388.20
Total		$146,307.09