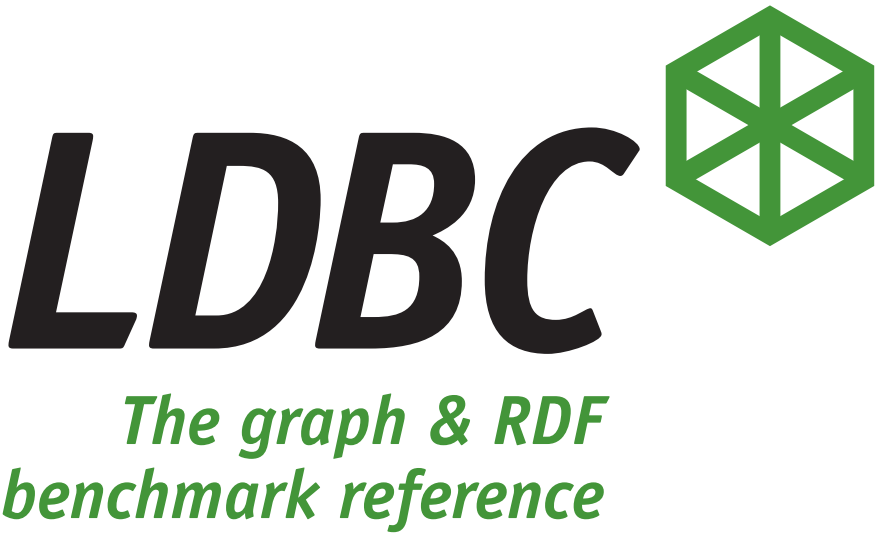

Full Disclosure Report of the LDBC Social Network Benchmark

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

October 5, 2024

General Terms

Executive Summary

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

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). TigerGraph previously passed the LDBC SNB BI benchmark with a single-machine on-premise setup using the SF1 000 dataset and with a multi-machine setup on AWS using the SF100, SF1 000 and SF10 000 datasets.

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 and its associated execution costs (compute, storage) were funded by AMD. This arrangement was deemed acceptable by both parties, the LDBC Board of Directors and the Auditor.

DocuSigned by:
David Puroja
C7FE4F71E2564AE...

10/7/2024

Mr. David Püroja
(Auditor)

Date

Pometry
(Audit Company)

DocuSigned by:
Gabor Szarnyas
FA33C2C8CB944A5...

10/7/2024

Dr. Gábor Szárnyas
(Leader of LDBC SNB Task Force)

Date

DocuSigned by:
Mingxi Wu
D302A312743E43D...

10/9/2024

Dr. Mingxi Wu
(Test Sponsor Representative)

Date

Table of Contents

1 Benchmark Description

The audit was conducted in compliance with the Social Network Benchmark Business Intelligence workload'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	1.0.3	https://github.com/ldbc/ldbc_snb_bi/releases/tag/v1.0.3

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 a Dell PowerEdge R7725 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 and Location

Hardware provider	AMD
Common name of the item	PowerEdge R7725 Server
Operating system	Ubuntu 22.04.4 LTS

2.1.2 CPU details

The details below were obtained using the command `cat /proc/cpuinfo` (Listing A.1) and `dmidecode -t processor` (Listing A.3) issued from the machine instance.

Table 2.2: CPU details summary

Type	AMD® EPYC® 9355 CPU
Total number	2
Cores per CPU	32
Threads per CPU core	2
CPU clock frequency	3.545GHz
Total cache size per CPU	L1i cache: 2MiB L1d cache: 3MiB L2 cache: 64MiB L3 cache: 512MiB

2.1.3 Memory details

The total size of the memory installed is 1.5TB and the type of memory is DDR5 with frequency 6400MHz. This information was obtained using the `cat /proc/meminfo` (Listing A.4) and `lshw -c memory` command (Listing A.5).

2.1.4 Disk and storage details

The instance used three NVMe SSDs. The SSDs used by the SUT and to store the data were formatted in `xfs`, the SSD used for the system was formatted in `ext4`.

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 44 525 IOPS.

Table 2.3: Disk details summary

Root disk	Dell NVMe PM1743 RI E3.S 3.84TB
Data disk	Dell NVMe PM1743 RI E3.S 3.84TB
Root file system	ext4
Data file system	xfs

2.1.5 Network details

The benchmark run only used a single machine, therefore no network details are included.

2.1.6 Machine pricing

The system pricing summary in US dollars (USD) is included in the table below. The full breakdown of the cost for the Dell PowerEdge R7725 Server, provided by AMD, can be found in Appendix E.

Table 2.4: Pricing summary

Item	**Price**
Dell PowerEdge R7725 Server	191 461.32 USD
Software license (3 years)	1 097 280.00 USD
Maintenance fee (3 years)	109 728.00 USD
Total cost	1 398 469.32 USD

2.1.7 System version and availability

Table 2.5: System versions

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

Note, for this audit, the installation script for 3.7.0 was edited to add Ubuntu 22.04 as supported operating system. All required installation files are included in the supplemental package of this audit.

2.1.8 Additional configuration

For the benchmark, additional configuration parameters were configured for the Tigergraph instance (executed on the machine instance CLI:

- gadmin config set FileLoader.ReplicaNumber 8
- gadmin config set FileLoader.Factory.HandlerCount 8
- gadmin config set RESTPP.Factory.HandlerCount 8
- gadmin config set KafkaLoader.Factory.HandlerCount 8
- gadmin config entry GPE.BasicConfig.Env, added MVExtraCopy=0;

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)
Data format for Umbra	composite-merged-fk layout, compressed CSV files
Scale factors for Umbra	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-merged-fk.tar.zst`
- `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 setup time consisted of the schema setup, query installations, precompute of auxiliary data structures (see Section 4.2) and rebuilding the indices. The column **Data loading time** shows how long it took to create a graph from the input CSV files and perform the initial indexing, compilation of the queries and precomputation. The column **Total time** contains the sum of the data preprocessing and loading times.

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

The decomposition of the **Data loading times** are shown in Table 3.3.

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

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

Scale factor	**Setup time (s)**	**Data loading time (s)**	**Total time (s)**
1000	902	7 804	8 706

Table 3.3: Decomposition of data loading time

	Time (s)
Schema setup	39
Load data	7 804
Query install	48
Precompute	236
Rebuild	579
Total	8 706

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.

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

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=/tg/r2/sf${SF}
# export TG_PARAMETER=/home/tigergraph/ldbc_snb_bi/parameters-sf${SF}

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

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/$
167.57 minutes	34 949.74	24.99	15 089.13	10.79

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

Query	Count	Min.	Max.	Mean	P_{50}	P_{90}	P_{95}	P_{99}
1	30	6.133	7.859	6.352	6.293	6.431	6.534	7.859
2a	30	1.418	16.430	6.331	4.157	13.382	14.211	16.430
2b	30	0.599	3.214	1.610	1.376	2.448	3.197	3.214
3	30	2.195	8.010	3.668	2.950	6.306	6.780	8.010
4	30	1.321	1.497	1.378	1.368	1.414	1.453	1.497
5	30	1.037	1.088	1.066	1.067	1.080	1.083	1.088
6	30	0.849	0.906	0.883	0.885	0.901	0.904	0.906
7	30	1.770	1.946	1.851	1.843	1.900	1.937	1.946
8a	30	1.531	2.380	1.755	1.705	1.890	2.332	2.380
8b	30	0.828	0.915	0.854	0.851	0.894	0.905	0.915
9	30	4.771	5.129	4.967	4.973	5.093	5.114	5.129
10a	30	3.847	7.661	5.500	5.603	6.378	6.498	7.661
10b	30	2.220	2.929	2.642	2.705	2.869	2.909	2.929
11	30	2.669	3.141	2.953	2.960	3.120	3.127	3.141
12	30	4.372	7.514	5.874	6.605	7.180	7.449	7.514
13	30	11.073	11.660	11.318	11.297	11.491	11.559	11.660
14a	30	6.446	7.454	7.118	7.106	7.296	7.330	7.454
14b	30	2.797	2.947	2.834	2.828	2.847	2.913	2.947
15a	30	10.843	13.339	11.566	11.550	12.015	12.083	13.339
15b	30	8.269	38.269	23.704	22.996	34.015	36.290	38.269
16a	30	3.751	9.750	5.500	5.022	6.842	7.139	9.750
16b	30	1.435	2.929	1.838	1.538	2.705	2.799	2.929
17	30	2.383	2.663	2.554	2.562	2.616	2.635	2.663
18	30	9.158	11.576	9.960	9.803	10.763	11.008	11.576
19a	30	1.956	2.897	2.586	2.692	2.870	2.878	2.897
19b	30	2.054	3.561	2.735	2.701	2.968	3.271	3.561
20a	30	0.411	2.454	1.037	0.417	2.251	2.449	2.454
20b	30	0.614	1.223	0.845	0.817	1.023	1.028	1.223

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
reads	3 938.587
writes	966.927
q4precomputation	51.865
q6precomputation	53.580
q19precomputation	464.228
q20precomputation	18.111

6 Validation of the Results

The results were cross-validated against the Umbra version `cbad59200` [1] and Neo4j Community Edition version `4.4.13` [2] reference implementation on scale factor 10. Umbra is a relational database management system developed at the Technische Universität München. The queries of the BI workload are implemented using PostgreSQL-compatible SQL queries that use the `WITH RECURSIVE` clause to implement graph traversal operations. Neo4j is a graph database management system where data is stored using the property graph data model and is queried using the Cypher language.

Listing 6.1: Output of the cross-validation commands

```
$ 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

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

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

[1] https://github.com/ldbc/ldbc_snb_bi/tree/e84aadab0347e90da655f2dd5df8254012e63339/umbra

[2] https://github.com/ldbc/ldbc_snb_bi/tree/e84aadab0347e90da655f2dd5df8254012e63339/cypher

7 Supplementary Materials

Table 7.1: Supplementary materials

File or Directory	Purpose
`umbra/output/output-sf10`	Output of the Umbra reference implementation
`cypher/output/output-sf10`	Output of the Cypher 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-1.0.3.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
`tigergraph-3.7.0-offline.tar.gz`	Installer of TigerGraph

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

```
ldbc_snb_bi_tigergraph_sf1000_attachments
├── umbra
│   └── output
│       └── output-sf10
│           └── results.csv
├── 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
│       │   ├── results.csv
│       │   └── timings.csv
│       └── output-sf1000
│           ├── benchmark.csv
│           ├── load.csv
│           ├── results.csv
│           └── timings.csv
├── ldbc_snb_bi-1.0.3.tar.gz
├── ldbc_snb_datagen_spark-0.5.1.tar.gz
├── ldbc_snb_specification-2.2.0.pdf
└── tigergraph-3.7.0-offline.tar.gz
```

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  : 26
model               : 2
model name  : AMD EPYC 9355 32-Core Processor
stepping    : 1
microcode   : 0xb002114
cpu MHz             : 1736.316
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 amd_lbr_v2 nopl
 nonstop_tsc cpuid extd_apicid aperfmperf rapl 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 3
 dnowprefetch osvw ibs skinit wdt tce topoext perfctr_core perfctr_nb bpext perfctr_llc mwaitx cpb cat_l3
 cdp_l3 hw_pstate ssbd mba perfmon_v2 ibrs ibpb stibp ibrs_enhanced vmmcall fsgsbase tsc_adjust 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 user_shstk
  avx_vnni avx512_bf16 clzero irperf xsaveerptr rdpru wbnoinvd amd_ppin cppc arat npt lbrv svm_lock nrip_save
 tsc_scale vmcb_clean flushbyasid decodeassists pausefilter pfthreshold avic v_vmsave_vmload vgif x2avic
 v_spec_ctrl vnmi avx512vbmi umip pku ospke avx512_vbmi2 gfni vaes vpclmulqdq avx512_vnni avx512_bitalg
 avx512_vpopcntdq la57 rdpid bus_lock_detect movdiri movdir64b overflow_recov succor smca fsrm
 avx512_vp2intersect flush_l1d debug_swap
bugs                : sysret_ss_attrs spectre_v1 spectre_v2 spec_store_bypass
bogomips    : 7090.55
TLB size    : 192 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
CPU(s):              32
On-line CPU(s) list: 0-31
Thread(s) per core:  2
Core(s) per socket:  16
Socket(s):           1
NUMA node(s):        2
Vendor ID:           AuthenticAMD
CPU family:          25
Model:               1
```

```
    Model name:          AMD EPYC 7R13 Processor
    Stepping:            1
    CPU MHz:             1502.144
    BogoMIPS:            5299.97
    Hypervisor vendor:   KVM
    Virtualization type: full
    L1d cache:           32K
    L1i cache:           32K
    L2 cache:            512K
    L3 cache:            32768K
    NUMA node0 CPU(s):   0-7,16-23
    NUMA node1 CPU(s):   8-15,24-31
    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 tsc_known_freq pni pclmulqdq ssse3 fma cx16 pcid sse4_1 sse4_2 x2apic movbe popcnt aes
      xsave avx f16c rdrand hypervisor lahf_lm cmp_legacy cr8_legacy abm sse4a misalignsse 3dnowprefetch topoext
     invpcid_single ssbd ibrs ibpb stibp vmmcall fsgsbase bmi1 avx2 smep bmi2 erms invpcid rdseed adx smap
     clflushopt clwb sha_ni xsaveopt xsavec xgetbv1 clzero xsaveerptr rdpru wbnoinvd arat npt nrip_save vaes
     vpclmulqdq rdpid
```

Listing A.3: Output of the `dmidecode -t processor` command

```
Architecture:                    x86_64
CPU op-mode(s):                  32-bit, 64-bit
Address sizes:                   52 bits physical, 57 bits virtual
Byte Order:                      Little Endian
CPU(s):                          128
On-line CPU(s) list:             0-127
Vendor ID:                       AuthenticAMD
Model name:                      AMD EPYC 9355 32-Core Processor
CPU family:                      26
Model:                           2
Thread(s) per core:              2
Core(s) per socket:              32
Socket(s):                       2
Stepping:                        1
BogoMIPS:                        7090.55
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 amd_lbr_v2
     nopl nonstop_tsc cpuid extd_apicid aperfmperf rapl 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 hw_pstate ssbd mba perfmon_v2 ibrs ibpb stibp ibrs_enhanced vmmcall fsgsbase tsc_adjust 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 user_shstk
      avx_vnni avx512_bf16 clzero irperf xsaveerptr rdpru wbnoinvd amd_ppin cppc arat npt lbrv svm_lock nrip_save
     tsc_scale vmcb_clean flushbyasid decodeassists pausefilter pfthreshold avic v_vmsave_vmload vgif x2avic
     v_spec_ctrl vnmi avx512vbmi umip pku ospke avx512_vbmi2 gfni vaes vpclmulqdq avx512_vnni avx512_bitalg
     avx512_vpopcntdq la57 rdpid bus_lock_detect movdiri movdir64b overflow_recov succor smca fsrm
     avx512_vp2intersect flush_l1d debug_swap
Virtualization:                  AMD-V
L1d cache:                       3 MiB (64 instances)
L1i cache:                       2 MiB (64 instances)
L2 cache:                        64 MiB (64 instances)
L3 cache:                        512 MiB (16 instances)
NUMA node(s):                    2
NUMA node0 CPU(s):               0-31,64-95
NUMA node1 CPU(s):               32-63,96-127
```

```
Vulnerability Gather data sampling:   Not affected
Vulnerability Itlb multihit:          Not affected
Vulnerability L1tf:                   Not affected
Vulnerability Mds:                    Not affected
Vulnerability Meltdown:               Not affected
Vulnerability Mmio stale data:        Not affected
Vulnerability Reg file data sampling: Not affected
Vulnerability Retbleed:               Not affected
Vulnerability Spec rstack overflow:   Not affected
Vulnerability Spec store bypass:      Mitigation; Speculative Store Bypass disabled via prctl
Vulnerability Spectre v1:             Mitigation; usercopy/swapgs barriers and __user pointer sanitization
Vulnerability Spectre v2:             Mitigation; Enhanced / Automatic IBRS; IBPB conditional; STIBP always-on;
     RSB filling; PBRSB-eIBRS Not affected; BHI Not affected
Vulnerability Srbds:                  Not affected
Vulnerability Tsx async abort:        Not affected
```

Listing A.4: Output of the `cat /proc/meminfo` command

```
MemTotal:       1584566504 kB
MemFree:        1558998080 kB
MemAvailable:   1569463816 kB
Buffers:          198180 kB
Cached:         16787336 kB
SwapCached:            0 kB
Active:          5230956 kB
Inactive:       15929488 kB
Active(anon):    4608216 kB
Inactive(anon):        0 kB
Active(file):     622740 kB
Inactive(file): 15929488 kB
Unevictable:        3088 kB
Mlocked:               0 kB
SwapTotal:       1998844 kB
SwapFree:        1998844 kB
Zswap:                 0 kB
Zswapped:              0 kB
Dirty:               400 kB
Writeback:             0 kB
AnonPages:       4178928 kB
Mapped:           647272 kB
Shmem:            434356 kB
KReclaimable:     548120 kB
Slab:            1521800 kB
SReclaimable:     548120 kB
SUnreclaim:       973680 kB
KernelStack:       65520 kB
PageTables:        57692 kB
SecPageTables:         0 kB
NFS_Unstable:          0 kB
Bounce:                0 kB
WritebackTmp:          0 kB
CommitLimit:    794282096 kB
Committed_AS:   26829536 kB
VmallocTotal:   13743895347199 kB
VmallocUsed:      522408 kB
VmallocChunk:          0 kB
Percpu:           293376 kB
HardwareCorrupted:     0 kB
```

```
AnonHugePages:         0 kB
ShmemHugePages:        0 kB
ShmemPmdMapped:        0 kB
FileHugePages:         0 kB
FilePmdMapped:         0 kB
Unaccepted:            0 kB
HugePages_Total:       0
HugePages_Free:        0
HugePages_Rsvd:        0
HugePages_Surp:        0
Hugepagesize:       2048 kB
Hugetlb:               0 kB
DirectMap4k:      231916 kB
DirectMap2M:     9529344 kB
DirectMap1G:    1600126976 kB
```

Listing A.5: Output of the `lshw -C memory` command for a single memory bank

```
  *-memory
       description: System Memory
       physical id: 1000
       slot: System board or motherboard
       size: 1536GiB
       capacity: 6TiB
       capabilities: ecc
       configuration: errordetection=multi-bit-ecc
     *-bank:0
          description: DIMM Synchronous Registered (Buffered) 6400 MHz (0.2 ns)
          product: HMCG94AHBRA277N
          vendor: Hynix Semiconductor (Hyundai Electronics)
          physical id: 0
          serial: 97C7BF7C
          slot: A1
          size: 64GiB
          width: 64 bits
          clock: 2105MHz (0.5ns)
```

B IO PERFORMANCE

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

```
fio --rw=write --ioengine=sync --fdatasync=1 --direct=1 --directory=/tg-data/io-test-data --size=2g --bs=4k --
    name=iotest
iotest: (g=0): rw=write, bs=(R) 4096B-4096B, (W) 4096B-4096B, (T) 4096B-4096B, ioengine=sync, iodepth=1
fio-3.28
Starting 1 process
iotest: Laying out IO file (1 file / 2048MiB)
Jobs: 1 (f=1): [W(1)][100.0%][w=174MiB/s][w=44.7k IOPS][eta 00m:00s]
iotest: (groupid=0, jobs=1): err= 0: pid=49650: Thu Aug 29 20:53:24 2024
  write: IOPS=44.5k, BW=174MiB/s (182MB/s)(2048MiB/11774msec); 0 zone resets
    clat (nsec): min=8281, max=70856, avg=9021.47, stdev=684.82
     lat (nsec): min=8301, max=71517, avg=9050.05, stdev=685.96
    clat percentiles (nsec):
     |  1.00th=[ 8512],  5.00th=[ 8640], 10.00th=[ 8640], 20.00th=[ 8768],
     | 30.00th=[ 8768], 40.00th=[ 8896], 50.00th=[ 8896], 60.00th=[ 8896],
     | 70.00th=[ 9024], 80.00th=[ 9024], 90.00th=[ 9280], 95.00th=[10176],
     | 99.00th=[11456], 99.50th=[14528], 99.90th=[16064], 99.95th=[17024],
     | 99.99th=[21632]
   bw (  KiB/s): min=176096, max=179744, per=99.99%, avg=178102.26, stdev=802.88, samples=23
   iops        : min=44024, max=44936, avg=44525.57, stdev=200.72, samples=23
  lat (usec)   : 10=94.30%, 20=5.68%, 50=0.02%, 100=0.01%
  fsync/fdatasync/sync_file_range:
    sync (nsec): min=11065, max=168249, avg=13115.26, stdev=970.90
    sync percentiles (nsec):
     |  1.00th=[12096],  5.00th=[12352], 10.00th=[12480], 20.00th=[12608],
     | 30.00th=[12608], 40.00th=[12736], 50.00th=[12992], 60.00th=[13120],
     | 70.00th=[13248], 80.00th=[13376], 90.00th=[13888], 95.00th=[14400],
     | 99.00th=[17280], 99.50th=[18816], 99.90th=[21376], 99.95th=[23936],
     | 99.99th=[29568]
  cpu          : usr=1.16%, sys=31.80%, ctx=1568287, majf=0, minf=16
  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=174MiB/s (182MB/s), 174MiB/s-174MiB/s (182MB/s-182MB/s), io=2048MiB (2147MB), run=11774-11774msec

Disk stats (read/write):
  nvme1n1: ios=0/1035816, merge=0/3, ticks=0/6876, in_queue=6876, util=58.40%
```

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 was performed with the following command: `time find /home/tigergraph/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 Umbra 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);
```

```
47 }
48
49 RUN GLOBAL SCHEMA_CHANGE JOB addIndex
50 CREATE GRAPH ldbc_snb (*)
```

E Machine Cost Breakdown

Quantity	Description	Cost
1	Dell PowerEdge R7725 Server	$189,394.14
1	Chassis with up to 8 E3.S NVMe Direct Drives	$0.00
1	AMD EPYC 9355 3.40GHz, 32C/64T, 256 Cache (280W) DDR5-6000	$0.00
1	AMD EPYC 9355 3.40GHz, 32C/64T, 256 Cache (280W) DDR5-6000	$0.00
1	Heatsink for 2 CPU configuration (CPU less than or equal to 400W)	$0.00
1	Performance Optimized	$0.00
1	6400MT/s RDIMMs	$0.00
24	64GB RDIMM, 6400MT/s, Dual Rank	$0.00
1	C30, No RAID for NVME chassis	$0.00
1	No Controller	$0.00
1	No Hard Drive	$0.00
4	3.84TB NVMe Read Intensive AG Drive E3s Gen5 with carrier	$0.00
1	Performance BIOS Setting	$0.00
1	UEFI BIOS Boot Mode with GPT Partition	$0.00
1	PowerEdge 2U Very High Performance Fan	$0.00
1	Dual, Redundant (1+1),Hot-Plug MHS Power Supply, 1500W MM, Titanium	$0.00
2	NEMA 5-15P to C13 Wall Plug, 125 Volt, 15 AMP, 10 Feet (3m), Power Cord, North America	$0.00
1	Riser Config 3, 4 x16 FH Slots (Gen5), 2nd OCP	$0.00
1	PowerEdge R7725 Motherboard, MX	$0.00
1	Broadcom 57504 Quad Port 10/25GbE,SFP28, OCP NIC 3.0	$0.00
1	PowerEdge 2U Standard Bezel	$0.00
1	No Media Required	$0.00
1	iDRAC10, Enterprise 17G	$0.00
1	Blank Left Ear Module	$0.00
1	iDRAC Force Change Password for OCP cards	$0.00
1	iDRAC Service Module (ISM), NOT Installed	$0.00
1	ReadyRails Sliding Rails with Cable Management Arm	$0.00
1	PowerEdge R7725 Shipping	$0.00
1	PowerEdge 2U Shipping Material	$0.00
1	PowerEdge CCC, No CE Label Marking	$0.00
1	Decline Selection	$0.00
1	Realtek RTL8153 Gigabit Ethernet Adapter	$0.00
1	Basic Next Business Day 36 Months, 36 Month(s)	$249.00
1	ProSupport and Next Business Day Onsite Service, 36 Month(s)	$1,818.18
	Total	$191,461.32