Нагрузочное тестирование: Clickhouse

Тестирование производилось на следующем оборудовании:

Architecture: x86_64
CPU op-mode(s): 32-bit, 64-bit
Byte Order: Little Endian
Address sizes: 43 bits physical, 48 bits virtual
CPU(s): 16
On-line CPU(s) list: 0-15
Thread(s) per core: 2
Core(s) per socket: 8
Socket(s): 1
NUMA node(s): 1
Vendor ID: AuthenticAMD
CPU family: 23
Model: 8
Model name: AMD Ryzen 7 2700X Eight-Core Processor
Stepping: 2
CPU MHz: 3499.739
BogoMIPS: 6999.47
Virtualization: AMD-V
L1d cache: 256 KiB
L1i cache: 512 KiB
L2 cache: 4 MiB
L3 cache: 16 MiB

RAM: 64GB

DISK: NVMe 512GB

Тестирование производилось со следующим программным обеспечением:

Ubuntu 20.04.6 LTS Linux 5.15.0-89-generic

KVM QEMU emulator version 4.2.1

Docker version 24.0.5

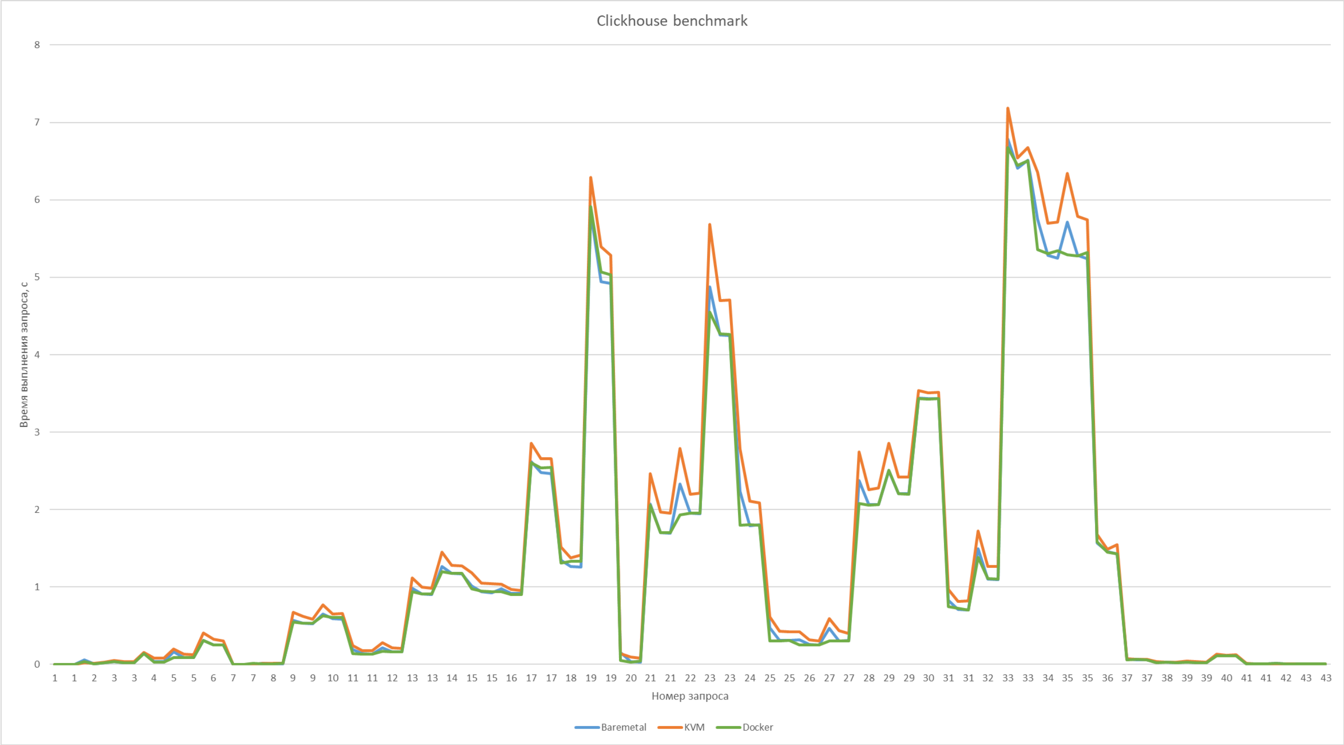
ClickHouse server version 23.11.1.2703

Используемые запросы:

1 SELECT count() FROM {table};
2 SELECT count() FROM {table} WHERE AdvEngineID != 0;
3 SELECT sum(AdvEngineID), count(), avg(ResolutionWidth) FROM {table} ;
4 SELECT sum(UserID) FROM {table} ;
5 SELECT uniq(UserID) FROM {table} ;
6 SELECT uniq(SearchPhrase) FROM {table} ;
7 SELECT min(EventDate), max(EventDate) FROM {table} ;
8 SELECT AdvEngineID, count() FROM {table} WHERE AdvEngineID != 0 GROUP BY AdvEngineID ORDER BY count() DESC;
9 SELECT RegionID, uniq(UserID) AS u FROM {table} GROUP BY RegionID ORDER BY u DESC LIMIT 10;
10 SELECT RegionID, sum(AdvEngineID), count() AS c, avg(ResolutionWidth), uniq(UserID) FROM {table} GROUP BY RegionID ORDER BY c DESC LIMIT 10;
11 SELECT MobilePhoneModel, uniq(UserID) AS u FROM {table} WHERE MobilePhoneModel != '' GROUP BY MobilePhoneModel ORDER BY u DESC LIMIT 10;
12 SELECT MobilePhone, MobilePhoneModel, uniq(UserID) AS u FROM {table} WHERE MobilePhoneModel != '' GROUP BY MobilePhone, MobilePhoneModel ORDER BY u DESC LIMIT 10;
13 SELECT SearchPhrase, count() AS c FROM {table} WHERE SearchPhrase != '' GROUP BY SearchPhrase ORDER BY c DESC LIMIT 10;
14 SELECT SearchPhrase, uniq(UserID) AS u FROM {table} WHERE SearchPhrase != '' GROUP BY SearchPhrase ORDER BY u DESC LIMIT 10;
15 SELECT SearchEngineID, SearchPhrase, count() AS c FROM {table} WHERE SearchPhrase != '' GROUP BY SearchEngineID, SearchPhrase ORDER BY c DESC LIMIT 10;
16 SELECT UserID, count() FROM {table} GROUP BY UserID ORDER BY count() DESC LIMIT 10;
17 SELECT UserID, SearchPhrase, count() FROM {table} GROUP BY UserID, SearchPhrase ORDER BY count() DESC LIMIT 10;

```
18 SELECT UserID, SearchPhrase, count() FROM {table} GROUP BY UserID, SearchPhrase LIMIT 10;
19 SELECT UserID, toMinute(EventTime) AS m, SearchPhrase, count() FROM {table} GROUP BY UserID, m, SearchPhrase ORDER BY count() DESC LIMIT 10;
20 SELECT UserID FROM {table} WHERE UserID = 12345678901234567890;
21 SELECT count() FROM {table} WHERE URL LIKE '%metrika%';
22 SELECT SearchPhrase, any(URL), count() AS c FROM {table} WHERE URL LIKE '%metrika%' AND SearchPhrase != '' GROUP BY SearchPhrase ORDER BY c DESC LIMIT 10;
23 SELECT SearchPhrase, any(URL), any(Title), count() AS c, uniq(UserID) FROM {table} WHERE Title LIKE '%Яндекс%' AND URL NOT LIKE '%.yandex.%' AND SearchPhrase != '' GROUP BY SearchPhrase ORDER BY c DESC LIMIT 10;
24 SELECT * FROM {table} WHERE URL LIKE '%metrika%' ORDER BY EventTime LIMIT 10;
25 SELECT SearchPhrase FROM {table} WHERE SearchPhrase != '' ORDER BY EventTime LIMIT 10;
26 SELECT SearchPhrase FROM {table} WHERE SearchPhrase != '' ORDER BY SearchPhrase LIMIT 10;
27 SELECT SearchPhrase FROM {table} WHERE SearchPhrase != '' ORDER BY EventTime, SearchPhrase LIMIT 10;
28 SELECT CounterID, avg(length(URL)) AS l, count() AS c FROM {table} WHERE URL != '' GROUP BY CounterID HAVING c > 100000 ORDER BY l DESC LIMIT 25;
29 SELECT domainWithoutWWW(Referer) AS key, avg(length(Referer)) AS l, count() AS c, any(Referer) FROM {table} WHERE Referer != '' GROUP BY key HAVING c > 100000 ORDER BY l DESC LIMIT 25;
30 SELECT sum(ResolutionWidth), sum(ResolutionWidth + 1), sum(ResolutionWidth + 2), sum(ResolutionWidth + 3), sum(ResolutionWidth + 4), sum(ResolutionWidth + 5), sum(ResolutionWidth + 6), sum(ResolutionWidth + 7), sum(ResolutionWidth + 8), sum(ResolutionWidth + 9), sum(ResolutionWidth + 10), sum(ResolutionWidth + 11), sum(ResolutionWidth + 12), sum(ResolutionWidth + 13), sum(ResolutionWidth + 14), sum(ResolutionWidth + 15), sum(ResolutionWidth + 16), sum(ResolutionWidth + 17), sum(ResolutionWidth + 18), sum(ResolutionWidth + 19), sum(ResolutionWidth + 20), sum(ResolutionWidth + 21), sum(ResolutionWidth + 22), sum(ResolutionWidth + 23), sum(ResolutionWidth + 24), sum(ResolutionWidth + 25), sum(ResolutionWidth + 26), sum(ResolutionWidth + 27), sum(ResolutionWidth + 28), sum(ResolutionWidth + 29), sum(ResolutionWidth + 30), sum(ResolutionWidth + 31), sum(ResolutionWidth + 32), sum(ResolutionWidth + 33), sum(ResolutionWidth + 34), sum(ResolutionWidth + 35), sum(ResolutionWidth + 36), sum(ResolutionWidth + 37), sum(ResolutionWidth + 38), sum(ResolutionWidth + 39), sum(ResolutionWidth + 40), sum(ResolutionWidth + 41), sum(ResolutionWidth + 42), sum(ResolutionWidth + 43), sum(ResolutionWidth + 44), sum(ResolutionWidth + 45), sum(ResolutionWidth + 46), sum(ResolutionWidth + 47), sum(ResolutionWidth + 48), sum(ResolutionWidth + 49), sum(ResolutionWidth + 50), sum(ResolutionWidth + 51), sum(ResolutionWidth + 52), sum(ResolutionWidth + 53), sum(ResolutionWidth + 54), sum(ResolutionWidth + 55), sum(ResolutionWidth + 56), sum(ResolutionWidth + 57), sum(ResolutionWidth + 58), sum(ResolutionWidth + 59), sum(ResolutionWidth + 60), sum(ResolutionWidth + 61), sum(ResolutionWidth + 62), sum(ResolutionWidth + 63), sum(ResolutionWidth + 64), sum(ResolutionWidth + 65), sum(ResolutionWidth + 66), sum(ResolutionWidth + 67), sum(ResolutionWidth + 68), sum(ResolutionWidth + 69), sum(ResolutionWidth + 70), sum(ResolutionWidth + 71), sum(ResolutionWidth + 72), sum(ResolutionWidth + 73), sum(ResolutionWidth + 74), sum(ResolutionWidth + 75), sum(ResolutionWidth + 76), sum(ResolutionWidth + 77), sum(ResolutionWidth + 78), sum(ResolutionWidth + 79), sum(ResolutionWidth + 80), sum(ResolutionWidth + 81), sum(ResolutionWidth + 82), sum(ResolutionWidth + 83), sum(ResolutionWidth + 84), sum(ResolutionWidth + 85), sum(ResolutionWidth + 86), sum(ResolutionWidth + 87), sum(ResolutionWidth + 88), sum(ResolutionWidth + 89) FROM {table};
31 SELECT SearchEngineID, ClientIP, count() AS c, sum(Refresh), avg(ResolutionWidth) FROM {table} WHERE SearchPhrase != '' GROUP BY SearchEngineID, ClientIP ORDER BY c DESC LIMIT 10;
32 SELECT WatchID, ClientIP, count() AS c, sum(Refresh), avg(ResolutionWidth) FROM {table} WHERE SearchPhrase != '' GROUP BY WatchID, ClientIP ORDER BY c DESC LIMIT 10;
33 SELECT WatchID, ClientIP, count() AS c, sum(Refresh), avg(ResolutionWidth) FROM {table} GROUP BY WatchID, ClientIP ORDER BY c DESC LIMIT 10;
34 SELECT URL, count() AS c FROM {table} GROUP BY URL ORDER BY c DESC LIMIT 10;
35 SELECT 1, URL, count() AS c FROM {table} GROUP BY 1, URL ORDER BY c DESC LIMIT 10;
36 SELECT ClientIP AS x, x - 1, x - 2, x - 3, count() AS c FROM {table} GROUP BY x, x - 1, x - 2, x - 3 ORDER BY c DESC LIMIT 10;
37 SELECT URL, count() AS PageViews FROM {table} WHERE CounterID = 62 AND EventDate >= '2013-07-01' AND EventDate <= '2013-07-31' AND NOT DontCountHits AND NOT Refresh AND notEmpty(URL) GROUP BY URL ORDER BY PageViews DESC LIMIT 10;
38 SELECT Title, count() AS PageViews FROM {table} WHERE CounterID = 62 AND EventDate >= '2013-07-01' AND EventDate <= '2013-07-31' AND NOT DontCountHits AND NOT Refresh AND notEmpty(Title) GROUP BY Title ORDER BY PageViews DESC LIMIT 10;
39 SELECT URL, count() AS PageViews FROM {table} WHERE CounterID = 62 AND EventDate >= '2013-07-01' AND EventDate <= '2013-07-31' AND NOT Refresh AND IsLink AND NOT IsDownload GROUP BY URL ORDER BY PageViews DESC LIMIT 1000;
40 SELECT TraficSourceID, SearchEngineID, AdvEngineID, ((SearchEngineID = 0 AND AdvEngineID = 0) ? Referer : '') AS Src, URL AS Dst, count() AS PageViews FROM {table} WHERE CounterID = 62 AND EventDate >= '2013-07-01' AND EventDate <= '2013-07-31' AND NOT Refresh GROUP BY TraficSourceID, SearchEngineID, AdvEngineID, Src, Dst ORDER BY PageViews DESC LIMIT 1000;
41 SELECT URLHash, EventDate, count() AS PageViews FROM {table} WHERE CounterID = 62 AND EventDate >= '2013-07-01' AND EventDate <= '2013-07-31' AND NOT Refresh AND TraficSourceID IN (-1, 6) AND RefererHash = halfMD5('http://example.ru/') GROUP BY URLHash, EventDate ORDER BY PageViews DESC LIMIT 100;
42 SELECT WindowClientWidth, WindowClientHeight, count() AS PageViews FROM {table} WHERE CounterID = 62 AND EventDate >= '2013-07-01' AND EventDate <= '2013-07-31' AND NOT Refresh AND NOT DontCountHits AND URLHash = halfMD5('http://example.ru/') GROUP BY WindowClientWidth, WindowClientHeight ORDER BY PageViews DESC LIMIT 10000;
43 SELECT toStartOfMinute(EventTime) AS Minute, count() AS PageViews FROM {table} WHERE CounterID = 62 AND EventDate >= '2013-07-01' AND EventDate <= '2013-07-02' AND NOT Refresh AND NOT DontCountHits GROUP BY Minute ORDER BY Minute;
```

Результаты тестирования:



	Baremetal	KVM	Docker
1	0,001	0,003	0,001
1	0,001	0,002	0,001
1	0,001	0,002	0,001
2	0,056	0,019	0,037
2	0,007	0,015	0,007
2	0,024	0,029	0,024
3	0,039	0,048	0,038
3	0,025	0,038	0,021
3	0,022	0,038	0,021
4	0,145	0,154	0,144
4	0,035	0,084	0,031
4	0,034	0,082	0,030
5	0,164	0,202	0,090
5	0,092	0,130	0,090
5	0,091	0,128	0,090
6	0,313	0,407	0,313
6	0,250	0,323	0,254
6	0,248	0,304	0,252
7	0,002	0,002	0,002
7	0,002	0,002	0,002
7	0,002	0,002	0,012
8	0,011	0,014	0,010

8	0,010	0,013	0,010
8	0,010	0,013	0,011
9	0,567	0,670	0,548
9	0,529	0,624	0,535
9	0,524	0,584	0,533
10	0,651	0,772	0,627
10	0,589	0,651	0,608
10	0,587	0,658	0,607
11	0,189	0,247	0,144
11	0,139	0,174	0,136
11	0,136	0,174	0,136
12	0,212	0,280	0,170
12	0,166	0,214	0,166
12	0,165	0,209	0,165
13	0,986	1,117	0,941
13	0,907	0,998	0,912
13	0,902	0,984	0,909
14	1,268	1,450	1,198
14	1,173	1,278	1,176
14	1,167	1,275	1,175
15	1,014	1,182	0,980
15	0,937	1,052	0,947
15	0,926	1,041	0,942
16	0,978	1,035	0,937
16	0,916	0,969	0,900
16	0,919	0,952	0,904
17	2,620	2,855	2,604
17	2,478	2,654	2,540
17	2,466	2,656	2,549
18	1,349	1,515	1,312
18	1,267	1,375	1,335
18	1,257	1,414	1,334
19	5,837	6,289	5,911
19	4,943	5,394	5,067
19	4,924	5,282	5,034
20	0,156	0,139	0,052
20	0,038	0,097	0,031
20	0,032	0,082	0,041
21	2,069	2,462	2,058
21	1,701	1,967	1,699
21	1,696	1,956	1,699
22	2,331	2,792	1,934

22	1,956	2,199	1,951
22	1,947	2,215	1,954
23	4,879	5,683	4,549
23	4,254	4,700	4,273
23	4,245	4,710	4,266
24	2,234	2,779	1,796
24	1,791	2,112	1,805
24	1,802	2,087	1,801
25	0,473	0,615	0,306
25	0,310	0,430	0,306
25	0,309	0,418	0,313
26	0,315	0,418	0,255
26	0,256	0,315	0,252
26	0,253	0,304	0,254
27	0,469	0,595	0,306
27	0,306	0,436	0,306
27	0,309	0,398	0,305
28	2,373	2,742	2,081
28	2,064	2,259	2,060
28	2,061	2,280	2,064
29	2,502	2,860	2,508
29	2,206	2,419	2,202
29	2,206	2,422	2,201
30	3,441	3,539	3,430
30	3,432	3,510	3,427
30	3,431	3,515	3,431
31	0,832	0,969	0,750
31	0,709	0,811	0,725
31	0,701	0,822	0,702
32	1,492	1,726	1,387
32	1,101	1,262	1,112
32	1,098	1,267	1,102
33	6,789	7,188	6,673
33	6,411	6,545	6,448
33	6,510	6,677	6,503
34	5,748	6,360	5,356
34	5,281	5,698	5,308
34	5,249	5,712	5,343
35	5,713	6,342	5,294
35	5,285	5,786	5,279
35	5,241	5,745	5,319
36	1,568	1,678	1,583

36	1,459	1,486	1,452
36	1,430	1,545	1,427
37	0,076	0,075	0,060
37	0,059	0,068	0,066
37	0,059	0,066	0,057
38	0,027	0,038	0,023
38	0,026	0,032	0,026
38	0,025	0,030	0,024
39	0,040	0,043	0,033
39	0,023	0,035	0,024
39	0,023	0,032	0,023
40	0,126	0,132	0,114
40	0,110	0,119	0,111
40	0,116	0,126	0,109
41	0,010	0,011	0,010
41	0,007	0,008	0,007
41	0,007	0,008	0,007
42	0,011	0,009	0,011
42	0,007	0,010	0,009
42	0,010	0,007	0,006
43	0,005	0,005	0,004
43	0,003	0,004	0,004
43	0,003	0,004	0,004