**Database info:**

- version: PostgreSQL 9.6.2
- size: 38GB
- rows in affected tables:
  - graphs – 98,266
  - graphs_items – 296,739
  - items – 586,273
  - hosts – 19,616
  - hosts_groups – 20,068
  - rights – 1,117

...and much smaller MySQL database, just to see query plans.

Typical graph.get requests looks as follow. I will use it to explain changes later.

```
SELECT … FROM … WHERE NOT EXISTS (SUBQUERY1) AND NOT EXISTS (SUBQUERY2) AND NOT EXISTS
(SUBQUERY3) AND other conditions;
```

**Possible improvements:**

- By limiting `SUBQUERY2` and `SUBQUERY3` it is possible gain improvement of speed (in my tests by ~15%, but it strongly depends on configuration of graphs).
  - Changes in *limit-subquery-2-and-3.patch*
- Joining `SUBQUERY2` and `SUBQUERY3` into a single query also gives small improvement, but much smaller than previous option (filtering out NULLs without subquery) and also conflicts with it. Improvement would be around 0.5% in execution speed.
  - Changes in *join-subquery-2-and-3-into-one-subquery.patch*
- Selecting host group in main query (instead of `SUBQUERY1`) gives improvement of speed of actual execution time by around 1/3. Memory consumption, as well as expected cost is around ½ lower. This comes with worse readability and risk of errors, but I was not able to get some.
  - Changes in *group-selection-in-main-query.patch*
- This is a little bit out of scope (since not an API problem), but if user has thousands of hosts and each if them has multiple graphs, graph drop-down filter generation in Monitoring → Graphs takes a lot of time. I worked with user which hsa permissions to 78K+ graphs. Before selecting Host group or Host, all graphs are listed in graph filter drop-down and printed on screen. CpageFilter::getGraphsCB() takes 2,5 seconds just to generate drop-down object and populate it with graphs (and rises to 4,5 seconds if order_result is changed to CArrayHelper::sort). It takes also multiple seconds for data transfer and HTML rendering. On the other hand, since most of hosts inherits it's graphs from templates, there are only 8 unique graph names (out of 78K). I do not see any reason why repeating graph names must be listed multiple times if user cannot distinguish one host's graph from others, without using host drop-down. I believe this can be done somehow smarter. Btw, this is also a case of ZBX-7706 (the query submitted in description do not have a `hostid`). An what is even worse – even super-admins cannot feel safe.

**What totaly doesn't work (I kept this list shorter to look more optimistic):**

- In my tests performance was decreased if accessibility of hosts or host groups was checked separately and added to query directly.

- Separately selecting all accessible graph items also gives much worse result in terms of speed. In my tests, graphitem.get worked very slow if called without specified graphids.
- Rewriting SUBQUERY1 in such a way that will select only rows that EXISTS (instead of NOT EXISTS) also gives almost no result. Estimated cost is a bit lower, but actual time, loops executed and memory was exact same for both approaches.