2023 W37 - Custom Plot API Request Duration

Topic	Custom Plot API Request Duration
Author	@Leonel Araújo @Pedro Cerejeira
Date	2023-09-11
Reviewers	@Hugo Fernandes @Pedro Cerejeira
Summary	Review constraints applied to Custom Plot functionality and analyze if they should be loosened. Custom Plot functionality performs raw-records-for-plot requests, which may be taking a long time due to middlewares such as “rate limiter” or “slowdown” and heavily impacting UX. After analyzing requests from 1st January 2023 until 5th September 2023, the main conclusions drawn are as follows: • As admin, on the edge cases, the amount of requests are not proportionally related to the delay; • As partner, 90% of the requests don’t take longer than 6 seconds. • The delays are most likely related to heavier requests, possibly related to variables not available to Partner, such as `battery_module_*`;
Follow-up	No actions needed.

🧵 Introduction

The Custom Plot functionality processes raw-records-for-plot requests, which can sometimes experience delays due to the presence of middlewares like the 'rate limiter' or 'slowdown,' adversely affecting the user experience. These middlewares ensure a sustainable load on `addvolt-api`.

Example of Custom Plot usage in Debug page.

This functionality is available to partners and admins on a Deal's "Troubleshooting" tab and to admins on "Debug" page. The requests performed by this functionality are limited to `raw-records-for-plot` , which is handled by `addvolt-api` and relayed to `addvolt-data-api`.

To gain a clearer understanding of request volumes and resolution times, we conducted an analysis on the amount of requests and the time they to take to resolve. We made a clear distinction between requests made by our partners and those made by administrators, as these requests are aimed at addressing different issues.

The period under analysis was from **1st January 2023** until **5th September 2023**.

🧪 Analysis

A. Requests per week day

To attain a deeper understanding of weekdays with the highest load i.e. the days with more requests. The total, p99, p90, max and avg metrics were calculated.

Admin

Weekday	total	p99	p90	max	avg
Monday	5523	679	293	**810**	162
Tuesday	5661	576	**329**	611	161

Wednesday	**6818**	**732**	274	807	**194**
Thursday	5283	498	238	530	150
Friday	4672	359	294	384	146
Saturday	209	39	38	40	17
Sunday	213	78	57	81	17

Values in bold are the highest value in each column. The main conclusion is that on weekends the amount of requests drops very significantly (only 10% of the requests are made over the weekend).

Partner

Weekday	total	p99	p90	max	avg
Monday	2670	322	172	368	83
Tuesday	2703	315	173	338	77
Wednesday	**3283**	**723**	139	857	**93**
Thursday	3269	584	**177**	778	93
Friday	2694	648	116	**885**	81
Saturday	169	63	44	66	16
Sunday	56	14	6	16	4

Values in bold are the highest value in each column. Again, requests drop significantly over the weekend.

B. Most active partner accounts

During our analysis, we questioned if the the most active partner accounts represented some edge case of some particularly problematic Deal:

Account	Name	Request Count	Company
765	Person A	21999	Client A
320	Person B	10179	Client B
305	Person C	8878	Client C

These value show the most active accounts, but are not necessarily pointing us to some problematic Deal or load problem.

C. Duration of highest requests count in a single day

The final analysis aimed to find specific days when an account made more requests. For each of these days, we will analyze the duration of the requests. Since we have rate limiter and slowdown middlewares to intentionally delay the duration, we are expecting that an account will wait longer the most requests they perform.

These should lead us to the edge cases and pinpoint the reports received from the users.

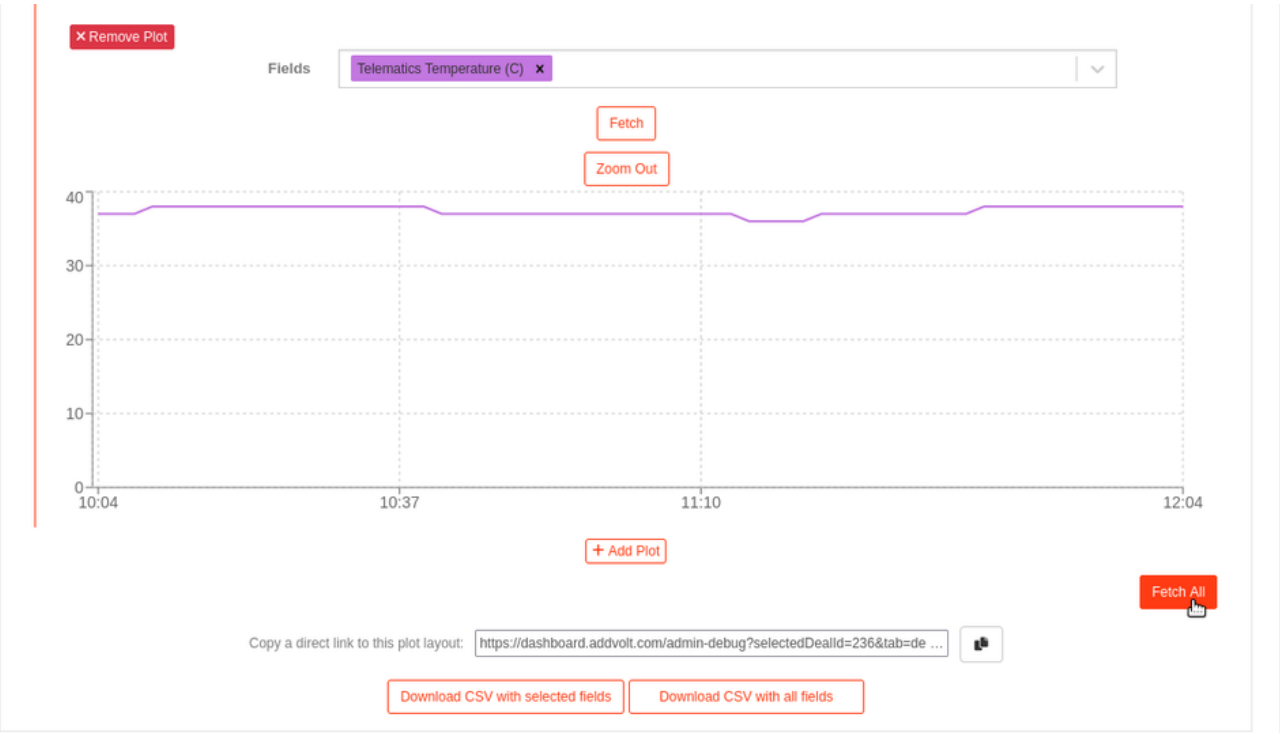
Admin

Day	Account	Request Count	sum	p99	p90	max	avg

2023-07-31[a]	306	**609**	**723,509**	6,210	1,531	31,881	1,188
2023-07-11	1054	337	122,076	1,642	580	2,609	362
2023-05-18[b]	14	330	**4,212,905**	31,033	19,904	32,520	12,766
2023-04-12	87	296	464,762	7,291	3,406	10,671	1,570
2023-06-13(c)	473	290	2,488,042	34,155	18,071	**60,408**	8,579
2023-04-12	44	258	324,519	4,125	2,732	5,473	1,257
2023-01-11	87	248	448,084	18,009	3,550	18,192	1,806
2023-05-22	1054	244	256,362	2,914	2,001	3,208	1,050
2023-05-23	1054	238	275,191	3,037	2,144	3,544	1,156
2023-05-23	44	225	414,385	1,4532	3,882	16,678	1,841

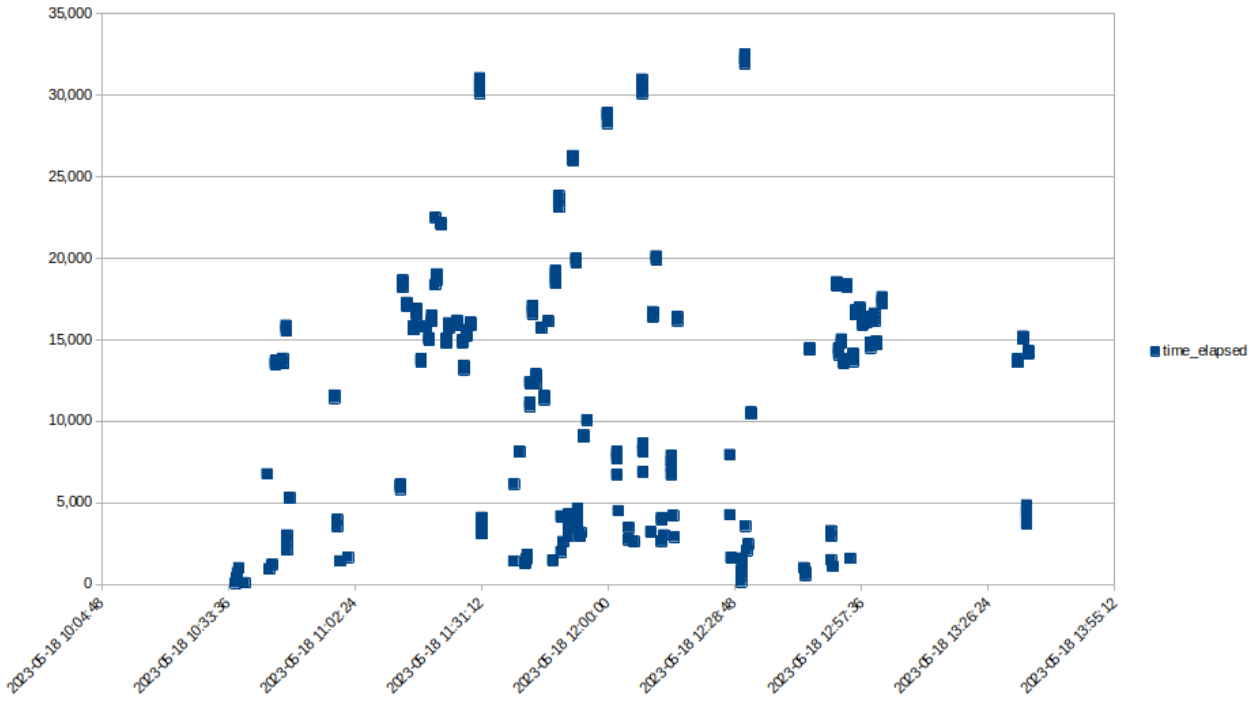
The most busy day (a), where account #306 performed **609** requests, waited in total 12 minutes (**723 seconds**) for plots alone on the 31st July. Still, this day does not represent the day with the most delayed responses.

More notably, on 18th May, the account #14 (b) waited 70 minutes (**4,212 seconds**). Upon a more thorough investigation of this case, it was possible to confirm the user was consistently populating 4 different plots and using the “Fetch All” functionality, which essentially performs requests 4 parallel requests.



Example “Fetch All” button that will perform request per created plot.

As such, it is very likely that the time waited by the user was in fact around 18 min (1/4 of the 70 minutes). Even though this is still a very high number, the wait was distributed over 2.5 hours of dashboard usage (10:30AM - 1PM).



An aggravating circumstance for this situation was that the data requested was from more than 4 months old. The rate limiter didn't seem to have major impact on the delay of these requests. The data of these requests is available in the attachments section. (1)

From this table, we can identify that at one point (c), account #473 had to wait **60 seconds** for a request to fulfill. Since the p99 for that day is 34 seconds, it's likely that the 60 second wait was a very rare case, and it most likely resulted in a "timeout" response.

All in all, as opposed to originally expected, the relationship between request count and request duration was not verified. **There are situations where a certain account had to wait a lot longer even when making fewer requests**.

Partner

Day	Account	Request Count	sum	p99	p90	max	avg
2023-03-23	605	588	795,651	7589	2637	10,396	1353
2023-03-22	605	565	673,675	5567	2403	6,947	1192
2023-03-24(d)	605	511	877,246	10513	3562	**14,485**	1716
2023-03-22	154	280	380,365	4806	3049	5,874	1358
2023-05-03	154	271	481,308	6552	3769	10,984	1776
2023-03-24	154	248	440,228	12264	3721	14,453	1775
2023-07-20	765	180	131,136	7313	889	7,461	728
2023-03-23	154	179	201,307	3995	2244	4,119	1124
2023-03-27 (e)	154	178	431,579	15924	6146	**17,519**	2424
2023-07-06	683	164	90,344	3864	1112	5,796	550

Even though the usage with Partner permissions is on the same scale in terms of amount of requests, the delays verified are at as severe. The highest delays were verified by account #605 (d) and #154 (e) but even in those cases, the duration of the requests is around **14-17 seconds**, which is not as high as the value noticed in Admin (where it would go up until 60 seconds).

It is worth noting that the great majority of requests (p90) are **not taking longer than 6 sec** at any point, which is still an okay response time for these types of requests.

👷 Conclusion

Main conclusions drawn, **out of the dataset chosen**:

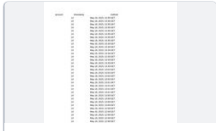
- As admin, on the edge cases, the amount of requests are not proportionally related to the delay;
- As partner, 90% of the requests don't take longer than 6 seconds.
- The delays are most likely related to heavier requests, possibly related to variables not available to Partner, such as `battery_module_*`;

For the next steps:

- No change to rate limiter or slowdown needed right now;

📎 Attachments

(1) Requests performed on 2023-05-18 by account #14



query_result_2… 1Z.xlsx
13 set. 2023, 01:18 PM