

**AT2-224 (Analysis of the performance of Webjive for loading complex dashboard)**  
**AT2-262 (Analysis of the performance of Webjive for refresh data on a complex dashboard)**

Before you start to read this document there are two concepts to keep in mind, what is Webjive loading and what is Webjive refreshing.

- **Webjive loading** is the time that webjive takes from the moment a user clicks the “start” button on the dashboard until the moment that all the widgets are visible on the dashboard and Webjive has sent all the subscription information to the backend (tangoGQL or the replacement daemon).
- **Webjive refreshing** is the step that comes after the loading and it’s defined by the time that webjive takes from the point that new data has arrived from the backend, until the point that all that data is updated and displayed on the dashboard. This takes into account the number of widgets on the dashboard, that is, it merges all the time necessary to update all the widgets on the screen.

This benchmark was done using with the original code of webjive, specifically the commit: <https://gitlab.com/MaxIV/webjive/tree/91472c124e9246b15d7462022b2d9c25a3a8cb79>  
The only change made to this was the backend point, this is, the URL from TangoGQL was changed to a local URL 127.0.0.1:1234 (WebjiveDaemon <https://gitlab.com/HFRibeiro/webjivedaemon> ), which sends the data in the same way that tangoGQL would send.

The overall results can be seen on the parent folder called *speedsite\_results*. This folder contains subfolders for each widget, with 1,10,100 and 1000 widgets each, in some cases webjive is crashing with 1000 widgets (AttributePlot and AttributeScatter) so only the 1, 10 and 100 are available.

The next analyses give a resumed summary for the loading and refresh times of each widget.

What speedsite gathers is the total loading time of the page itself, which means from the time you open the page on the browser to the time where you can actually interact with the page, but then some background functions kick in which are not well timed by speedsite. In fact, speedsite is not really capable of gathering real timing from the point you click the play button until the point the data really shows. (well it could be, but a major script development is necessary, out of scoop at this time)

That's why I've added the tables because for us and for future webjive use, those are the most important times.

Analysis of the performance of **Webjive for loading** complex dashboards.

- LED widgets:



Time in ms	1 Widget	10 Widgets	100 Widgets	1000 Widgets
Test1	18.1	25.6	53.9	326.5
Test2	22.4	25.4	47.4	298.6
Test3	20.8	24.7	48.5	294.6
Test4	16.1	19	47.1	291.7
Test5	17.4	22.7	46.8	300.9
Test6	15.8	34.2	50.6	272
Test7	26.6	21.1	47.1	272.4
Test8	18.5	21.2	46.9	278.3
Test9	17.1	22.8	48.4	285.9
Test10	16.9	22.7	49.6	287.1
<b>Mean Time (ms):</b>	<b>18.97</b>	<b>23.94</b>	<b>48.63</b>	<b>290.8</b>

- LED Widget sitespeed resumed results:

- 1 LED

Backend Time <b>8 ms</b> (13 ms)	Frontend Time <b>795 ms</b> (801 ms)	Page Load Time <b>803 ms</b> (810 ms)
-------------------------------------	---	--

- 10 LED

Backend Time <b>12 ms</b> (17 ms)	Frontend Time <b>833 ms</b> (864 ms)	Page Load Time <b>841 ms</b> (883 ms)
--------------------------------------	---	--

- 100 LED

Backend Time <b>9 ms</b> (13 ms)	Frontend Time <b>980 ms</b> (1,043 s)	Page Load Time <b>989 ms</b> (1,058 s)
-------------------------------------	--	---

- 1000 LED

Backend Time <b>8 ms</b> (9 ms)	Frontend Time <b>2.466 s</b> (2,628 s)	Page Load Time <b>2.475 s</b> (2,636 s)
------------------------------------	---	--

- **Attribute Display widgets:**

```
double_scalar: value
```

Time in ms	1 Widget	10 Widgets	100 Widgets	1000 Widgets
Test1	21.28	23.3	50.9	244.5
Test2	15.4	23.4	41.5	230.9
Test3	15.7	21.1	41.3	209.9
Test4	15.8	20.7	40.7	215.1
Test5	15.6	19.4	43.8	209.9
Test6	19.2	19.7	37.1	209.1
Test7	14.8	17.8	38.9	216.1
Test8	14.4	21.9	44.12	211.2
Test9	18.4	26.4	43.12	207.3
Test10	14.9	19	38.48	219.8
<b>Mean Time (ms):</b>	<b>16.548</b>	<b>21.27</b>	<b>41.992</b>	<b>217.38</b>

- **LED Widget overall sitespeed resumed results:**

- **1 Attribute Display**

Backend Time <b>13 ms</b> (13 ms)	Frontend Time <b>1.023 s</b> (1.079 s)	Page Load Time <b>1.038 s</b> (1.091 s)
--------------------------------------	---	--

- **10 Attribute Display**

Backend Time <b>11 ms</b> (12 ms)	Frontend Time <b>995 ms</b> (1.045 s)	Page Load Time <b>1.009 s</b> (1.058 s)
--------------------------------------	--	--

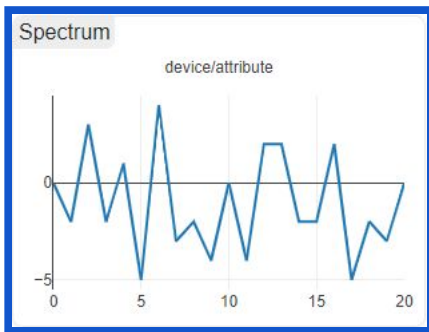
- **100 Attribute Display**

Backend Time <b>10 ms</b> (21 ms)	Frontend Time <b>1.092 s</b> (1.124 s)	Page Load Time <b>1.115 s</b> (1.136 s)
--------------------------------------	---	--

- **1000 Attribute Display**

Backend Time <b>13 ms</b> (14 ms)	Frontend Time <b>3.032 s</b> (3.216 s)	Page Load Time <b>3.048 s</b> (3.231 s)
--------------------------------------	---	--

- **Spectrum widgets:**



Time in ms	1 Widget	10 Widgets	100 Widgets	1000 Widgets
Test1	38.4	153.8	1380.9	23782.2
Test2	38.6	161.3	1673.6	23713.4
Test3	35.5	161.7	1525.5	25024.3
Test4	37.4	162.8	1428	24343.7
Test5	33.2	160.1	1453.5	25709.4
Test6	35.4	156.9	1489.3	24924.3
Test7	33.4	169.1	1592.5	24411.1
Test8	32.5	173.7	1536.5	25487.7
Test9	47.1	157.3	1449.2	25289.9
Test10	32.6	175.9	1525.3	25395.6
<b>Mean Time (ms):</b>	<b>36.41</b>	<b>163.26</b>	<b>1505.43</b>	<b>24808.16</b>

- **Spectrum Widget overall sitespeed resumed results:**

- **1 Spectrum**

Backend Time <b>12 ms</b> (13 ms)	Frontend Time <b>1.040 s</b> (1.162 s)	Page Load Time <b>1.053 s</b> (1.166 s)
--------------------------------------	---	--

- **10 Spectrum**

Backend Time <b>11 ms</b> (12 ms)	Frontend Time <b>1.049 s</b> (1.142 s)	Page Load Time <b>1.062 s</b> (1.153 s)
--------------------------------------	---	--

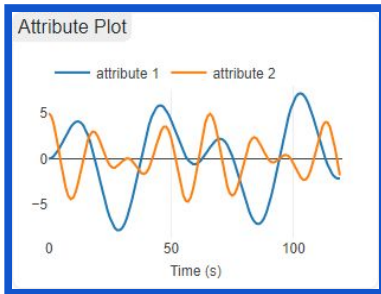
- **100 Spectrum**

Backend Time <b>12 ms</b> (12 ms)	Frontend Time <b>2.557 s</b> (2.669 s)	Page Load Time <b>2.570 s</b> (2.681 s)
--------------------------------------	---	--

- **1000 Spectrum**

Backend Time <b>10 ms</b> (12 ms)	Frontend Time <b>32.592 s</b> (36.454 s)	Page Load Time <b>32.604 s</b> (36.468 s)
--------------------------------------	---	--

- **Attribute plot widgets:**



Time in ms	1 Widget	10 Widgets	100 Widgets	1000 Widgets
Test1	57.4	377.1	12134.8	Crash
Test2	42.2	378.1	12298.3	Crash
Test3	43.8	421.2	12199.4	Crash
Test4	48.2	374.6	12252.8	Crash
Test5	58.4	369.9	12657.4	Crash
Test6	41.8	372	12502.6	Crash
Test7	41.1	372.5	12536.4	Crash
Test8	39.7	379.7	12489.6	Crash
Test9	41	385.7	12443.8	Crash
Test10	37.2	363	12697.4	Crash
<b>Mean Time (ms):</b>	<b>45.08</b>	<b>379.38</b>	<b>12421.25</b>	

- **Attribute plot overall sitespeed resumed results:**

- **1 Attribute plot**

Backend Time <b>17 ms</b> (22 ms)	Frontend Time <b>995 ms</b> (1.081 s)	Page Load Time <b>1.008 s</b> (1.098 s)
--------------------------------------	--	--

- **10 Attribute plot**

Backend Time <b>12 ms</b> (12 ms)	Frontend Time <b>1.198 s</b> (1.214 s)	Page Load Time <b>1.210 s</b> (1.229 s)
--------------------------------------	---	--

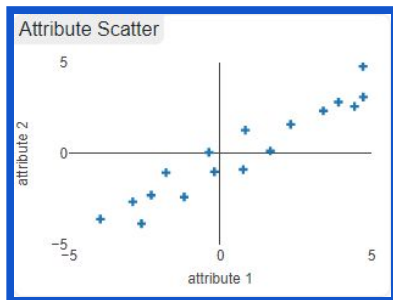
- **100 Attribute plot**

Backend Time <b>13 ms</b> (16 ms)	Frontend Time <b>3.191 s</b> (3.403 s)	Page Load Time <b>3.205 s</b> (3.417 s)
--------------------------------------	---	--

- **1000 Attribute plot**

Backend Time <b>CRASH</b>	Frontend Time <b>CRASH</b>	Page Load Time <b>CRASH</b>
------------------------------	-------------------------------	--------------------------------

- **Attribute scatter widgets:**



Time in ms	1 Widget	10 Widgets	100 Widgets	1000 Widgets
Test1	62	405.7	12606	Crash
Test2	50	383.6	12382.2	Crash
Test3	50.2	419.9	13033.5	Crash
Test4	49.8	394.2	12598.2	Crash
Test5	49.3	374.1	13110	Crash
Test6	44.4	412	12831.3	Crash
Test7	52.6	400.1	13289.9	Crash
Test8	44.3	390.6	13130.6	Crash
Test9	45.2	391.1	13489.6	Crash
Test10	50.1	387	13145.9	Crash
<b>Mean Time (ms):</b>	<b>49.79</b>	<b>395.83</b>	<b>12961.72</b>	

- **Attribute scatter overall sitespeed resumed results:**

- **1 Attribute scatter**

Backend Time <b>15 ms</b> (16 ms)	Frontend Time <b>1.099 s</b> (1.178 s)	Page Load Time <b>1.113 s</b> (1.195 s)
--------------------------------------	---	--

- **10 Attribute scatter**

Backend Time <b>12 ms</b> (15 ms)	Frontend Time <b>1.316 s</b> (1.328 s)	Page Load Time <b>1.334 s</b> (1.342 s)
--------------------------------------	---	--

- **100 Attribute scatter**

Backend Time <b>11 ms</b> (15 ms)	Frontend Time <b>2.903 s</b> (2.920 s)	Page Load Time <b>2.920 s</b> (2.933 s)
--------------------------------------	---	--

- **1000 Attribute scatter**

Backend Time <b>CRASH</b>	Frontend Time <b>CRASH</b>	Page Load Time <b>CRASH</b>
------------------------------	-------------------------------	--------------------------------

- **Attribute dial widgets:**



Time in ms	1 Widget	10 Widgets	100 Widgets	1000 Widgets
Test1	38.8	98.7	620.4	6388.1
Test2	29.6	96.1	844.6	6137.2
Test3	34	87.2	635.9	8135
Test4	40.3	84.6	589.4	7233.2
Test5	25.4	89	631.7	6437.6
Test6	26.5	78.8	645.7	7433.5
Test7	43.6	75.4	601	8238.4
Test8	26.2	81.2	609.5	7834.1
Test9	26.4	76	635.6	6234.2
Test10	28.1	82.9	611.8	7137.3
<b>Mean Time (ms):</b>	<b>31.89</b>	<b>84.99</b>	<b>642.56</b>	<b>7120.86</b>

- **Attribute dial overall sitespeed resumed results:**

- **1 Attribute dial**

Backend Time <b>10 ms</b> (18 ms)	Frontend Time <b>1.097 s</b> (1.125 s)	Page Load Time <b>1.118 s</b> (1.137 s)
--------------------------------------	---	--

- **10 Attribute dial**

Backend Time <b>12 ms</b> (14 ms)	Frontend Time <b>1.229 s</b> (1.380 s)	Page Load Time <b>1.243 s</b> (1.394 s)
--------------------------------------	---	--

- **100 Attribute dial**

Backend Time <b>8 ms</b> (15 ms)	Frontend Time <b>1.642 s</b> (1.697 s)	Page Load Time <b>1.651 s</b> (1.713 s)
-------------------------------------	---	--

- **1000 Attribute dial**

Backend Time <b>19 ms</b> (19 ms)	Frontend Time <b>8.235 s</b> (9.399 s)	Page Load Time <b>8.248 s</b> (9.424 s)
--------------------------------------	---	--

- **Logger widgets:**



Time in ms	1 Widget	10 Widgets	100 Widgets	1000 Widgets
Test1	33.7	44.5	197.4	3956.5
Test2	25	42.8	167.2	4560.8
Test3	21.9	42.8	180.2	3813.9
Test4	23	44.3	161.7	4740.7
Test5	23.7	40	204.4	4651.3
Test6	22.6	41.4	172.5	4829.7
Test7	20.4	37.6	171.8	3990.4
Test8	21.2	34.1	161.8	4813.1
Test9	20.5	36.3	160.5	4737.1
Test10	18.3	31.9	153.2	4741.2
<b>Mean Time (ms):</b>	<b>23.03</b>	<b>39.57</b>	<b>173.07</b>	<b>4483.47</b>

- **Logger overall sitespeed resumed results:**

- **1 Logger**

Backend Time <b>15 ms</b> (21 ms)	Frontend Time <b>1.011 s</b> (1.183 s)	Page Load Time <b>1.034 s</b> (1.204 s)
--------------------------------------	---	--

- **10 Logger**

Backend Time <b>9 ms</b> (10 ms)	Frontend Time <b>849 ms</b> (881 ms)	Page Load Time <b>858 ms</b> (892 ms)
-------------------------------------	---	--

- **100 Logger**

Backend Time <b>10 ms</b> (18 ms)	Frontend Time <b>1.031 s</b> (1.123 s)	Page Load Time <b>1.043 s</b> (1.143 s)
--------------------------------------	---	--

- **1000 Logger**

Backend Time <b>10 ms</b> (13 ms)	Frontend Time <b>2.031 s</b> (2.209 s)	Page Load Time <b>2.039 s</b> (2.220 s)
--------------------------------------	---	--



Analysis of the performance of **Webjive for refreshing** complex dashboards.

- **LED widgets:**



Time in ms	1 Widget	10 Widgets	100 Widgets	1000 Widgets
Test1	1.6	8.6	67.7	658.6
Test2	1.4	8.7	69.2	641.9
Test3	1.5	8.2	67	646.6
Test4	2.2	8.5	67.4	633.8
Test5	1.7	11.3	71.4	635.8
Test6	2.3	9.7	71.3	650.5
Test7	1.4	9.8	68.2	643.8
Test8	1.2	8.7	68.1	657.6
Test9	1.5	8.7	68.3	649.7
Test10	1.5	9.3	68.4	650.6
<b>Mean Time (ms):</b>	<b>1.63</b>	<b>9.15</b>	<b>68.7</b>	<b>646.89</b>

- **LED Widget sitespeed resumed results:**

- **1 LED**

Backend Time <b>8 ms</b> (13 ms)	Frontend Time <b>795 ms</b> (801 ms)	Page Load Time <b>803 ms</b> (810 ms)
-------------------------------------	---	--

- **10 LED**

Backend Time <b>12 ms</b> (17 ms)	Frontend Time <b>833 ms</b> (864 ms)	Page Load Time <b>841 ms</b> (883 ms)
--------------------------------------	---	--

- **100 LED**

Backend Time <b>9 ms</b> (13 ms)	Frontend Time <b>980 ms</b> (1,043 s)	Page Load Time <b>989 ms</b> (1,058 s)
-------------------------------------	--	---

- **1000 LED**

Backend Time <b>8 ms</b> (9 ms)	Frontend Time <b>2.466 s</b> (2,528 s)	Page Load Time <b>2.475 s</b> (2,536 s)
------------------------------------	---	--

- **Attribute Display widgets:**

```
double_scalar: value
```

Time in ms	1 Widget	10 Widgets	100 Widgets	1000 Widgets
Test1	1.3	12.9	64.1	464.4
Test2	1.3	10.6	45.2	434.6
Test3	1.3	10.7	47	430.7
Test4	3.3	12.2	46	436
Test5	1.5	11.6	50.2	450.3
Test6	1.7	10.3	47.9	448.9
Test7	1.7	10.2	49.3	445.9
Test8	1.4	9.8	46.9	428
Test9	1.5	9.7	51.5	428.3
Test10	2	9	49.8	454
<b>Mean Time (ms):</b>	<b>1.7</b>	<b>10.7</b>	<b>49.79</b>	<b>442.11</b>

- **LED Widget overall sitespeed resumed results:**

- **1 Attribute Display**

Backend Time <b>13 ms</b> (13 ms)	Frontend Time <b>1.023 s</b> (1.079 s)	Page Load Time <b>1.038 s</b> (1.091 s)
--------------------------------------	---	--

- **10 Attribute Display**

Backend Time <b>11 ms</b> (12 ms)	Frontend Time <b>995 ms</b> (1.045 s)	Page Load Time <b>1.009 s</b> (1.058 s)
--------------------------------------	--	--

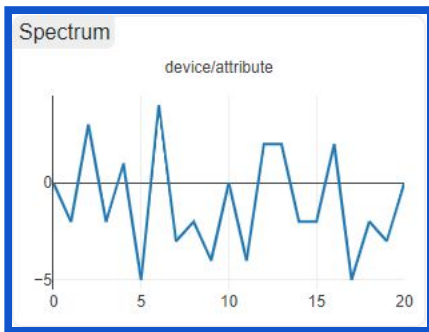
- **100 Attribute Display**

Backend Time <b>10 ms</b> (21 ms)	Frontend Time <b>1.092 s</b> (1.124 s)	Page Load Time <b>1.115 s</b> (1.136 s)
--------------------------------------	---	--

- **1000 Attribute Display**

Backend Time <b>13 ms</b> (14 ms)	Frontend Time <b>3.032 s</b> (3.216 s)	Page Load Time <b>3.048 s</b> (3.231 s)
--------------------------------------	---	--

- **Spectrum widgets:**



Time in ms	1 Widget	10 Widgets	100 Widgets	1000 Widgets
Test1	14	159.4	2397.3	65227
Test2	13	162.9	2232.5	52433.7
Test3	18.1	158.1	2073	47393.4
Test4	13.9	165.5	2155.5	52296.8
Test5	11.8	163.4	2202.4	48814.2
Test6	11.6	161.7	2289.6	55639.2
Test7	12.3	157.6	2150.3	55550.6
Test8	16.9	160.7	2341.5	51571.9
Test9	15.1	152.8	2313.3	49396.8
Test10	12.6	165.4	2919	49089.3
<b>Mean Time (ms):</b>	<b>13.93</b>	<b>160.75</b>	<b>2307.44</b>	<b>52741.29</b>

- **Spectrum Widget overall sitespeed resumed results:**

- **1 Spectrum**

Backend Time <b>12 ms</b> (13 ms)	Frontend Time <b>1.040 s</b> (1.162 s)	Page Load Time <b>1.053 s</b> (1.166 s)
--------------------------------------	---	--

- **10 Spectrum**

Backend Time <b>11 ms</b> (12 ms)	Frontend Time <b>1.049 s</b> (1.142 s)	Page Load Time <b>1.062 s</b> (1.153 s)
--------------------------------------	---	--

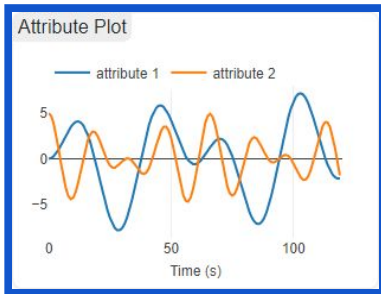
- **100 Spectrum**

Backend Time <b>12 ms</b> (12 ms)	Frontend Time <b>2.557 s</b> (2.669 s)	Page Load Time <b>2.570 s</b> (2.681 s)
--------------------------------------	---	--

- **1000 Spectrum**

Backend Time <b>10 ms</b> (12 ms)	Frontend Time <b>32.592 s</b> (36.454 s)	Page Load Time <b>32.604 s</b> (36.468 s)
--------------------------------------	---	--

- **Attribute plot widgets:**



Time in ms	1 Widget	10 Widgets	100 Widgets	1000 Widgets
Test1	39.9	1625.6	118499.1	Crash
Test2	33.6	1641.2	146706.8	Crash
Test3	35.8	1562	145445.3	Crash
Test4	34.5	1546.5	145926.4	Crash
Test5	38.2	1551.7	146718.6	Crash
Test6	33.8	1619.4	147495.8	Crash
Test7	37	1637.8	147107.4	Crash
Test8	37.1	1605.8	149510.3	Crash
Test9	32.9	1581	150325.8	Crash
Test10	32.4	1615.5	151341.1	Crash
<b>Mean Time (ms):</b>	<b>35.52</b>	<b>1598.65</b>	<b>144907.7</b>	

- **Attribute plot overall sitespeed resumed results:**

- **1 Attribute plot**

Backend Time <b>17 ms</b> (22 ms)	Frontend Time <b>995 ms</b> (1.081 s)	Page Load Time <b>1.008 s</b> (1.098 s)
--------------------------------------	--	--

- **10 Attribute plot**

Backend Time <b>12 ms</b> (12 ms)	Frontend Time <b>1.198 s</b> (1.214 s)	Page Load Time <b>1.210 s</b> (1.229 s)
--------------------------------------	---	--

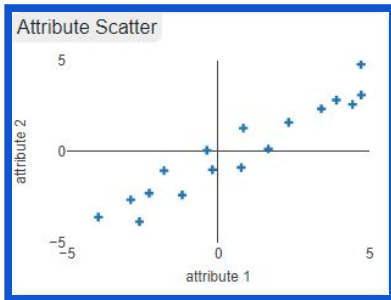
- **100 Attribute plot**

Backend Time <b>13 ms</b> (16 ms)	Frontend Time <b>3.191 s</b> (3.403 s)	Page Load Time <b>3.205 s</b> (3.417 s)
--------------------------------------	---	--

- **1000 Attribute plot**

Backend Time <b>CRASH</b>	Frontend Time <b>CRASH</b>	Page Load Time <b>CRASH</b>
------------------------------	-------------------------------	--------------------------------

- **Attribute scatter widgets:**



Time in ms	1 Widget	10 Widgets	100 Widgets	1000 Widgets
Test1	62.5	1914.7	111112.9	Crash
Test2	62.3	1960.3	137838.5	Crash
Test3	68.9	1962.1	137803.2	Crash
Test4	63.4	1886.4	137497.9	Crash
Test5	60.8	1925.7	137843.5	Crash
Test6	72.1	1852.5	137664.5	Crash
Test7	63.2	1883.5	137674.3	Crash
Test8	65	1893.4	137796.4	Crash
Test9	61.5	1885.4	138813.9	Crash
Test10	62.6	1880	140083.9	Crash
<b>Mean Time (ms):</b>	<b>64.23</b>	<b>1904.4</b>	<b>135412.9</b>	

- **Attribute scatter overall sitespeed resumed results:**

- **1 Attribute scatter**

Backend Time <b>15 ms</b> (16 ms)	Frontend Time <b>1.099 s</b> (1.178 s)	Page Load Time <b>1.113 s</b> (1.195 s)
--------------------------------------	---	--

- **10 Attribute scatter**

Backend Time <b>12 ms</b> (15 ms)	Frontend Time <b>1.316 s</b> (1.328 s)	Page Load Time <b>1.334 s</b> (1.342 s)
--------------------------------------	---	--

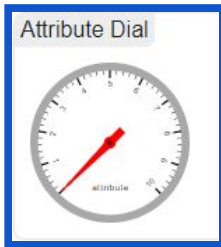
- **100 Attribute scatter**

Backend Time <b>11 ms</b> (15 ms)	Frontend Time <b>2.903 s</b> (2.920 s)	Page Load Time <b>2.920 s</b> (2.933 s)
--------------------------------------	---	--

- **1000 Attribute scatter**

Backend Time <b>CRASH</b>	Frontend Time <b>CRASH</b>	Page Load Time <b>CRASH</b>
------------------------------	-------------------------------	--------------------------------

- **Attribute dial widgets:**



Time in ms	1 Widget	10 Widgets	100 Widgets	1000 Widgets
Test1	6.1	65	711.5	8494.4
Test2	5.1	65.3	756.3	9421.7
Test3	4.9	64.1	748.2	8098.7
Test4	6.9	70.2	850.2	8642.6
Test5	5	63.7	772.9	7927.4
Test6	6.8	70.4	732.1	8590
Test7	4.4	66.5	730.8	7802.5
Test8	4.5	74.7	846.3	8669.8
Test9	4.5	63.7	823.4	7860.9
Test10	5.9	65.8	726.1	8576
<b>Mean Time (ms):</b>	<b>5.41</b>	<b>66.94</b>	<b>769.78</b>	<b>8408.4</b>

- **Attribute dial overall sitespeed resumed results:**

- **1 Attribute dial**

Backend Time <b>10 ms</b> (18 ms)	Frontend Time <b>1.097 s</b> (1.125 s)	Page Load Time <b>1.118 s</b> (1.137 s)
--------------------------------------	---	--

- **10 Attribute dial**

Backend Time <b>12 ms</b> (14 ms)	Frontend Time <b>1.229 s</b> (1.380 s)	Page Load Time <b>1.243 s</b> (1.394 s)
--------------------------------------	---	--

- **100 Attribute dial**

Backend Time <b>8 ms</b> (15 ms)	Frontend Time <b>1.642 s</b> (1.697 s)	Page Load Time <b>1.651 s</b> (1.713 s)
-------------------------------------	---	--

- **1000 Attribute dial**

Backend Time <b>19 ms</b> (19 ms)	Frontend Time <b>8.235 s</b> (9.399 s)	Page Load Time <b>8.248 s</b> (9.424 s)
--------------------------------------	---	--

- **Logger widgets:**



Time in ms	1 Widget	10 Widgets	100 Widgets	1000 Widgets
Test1	6	53.5	385.9	3615
Test2	8.3	45.5	400.5	3948.7
Test3	8.3	47.7	420.5	3961.3
Test4	5.9	46.5	437.8	4384.2
Test5	6.7	48.3	493.9	4480.6
Test6	7.7	47.8	494.5	4732.2
Test7	9.3	45.6	485.6	5167.8
Test8	6.4	54.2	524.2	5107.3
Test9	8.5	62.4	530.2	5658.9
Test10	7.1	53.1	552.9	5673.5
<b>Mean Time (ms):</b>	<b>7.42</b>	<b>50.46</b>	<b>472.6</b>	<b>4672.95</b>

- **Logger overall sitespeed resumed results:**

- **1 Logger**

Backend Time <b>15 ms</b> (21 ms)	Frontend Time <b>1.011 s</b> (1.183 s)	Page Load Time <b>1.034 s</b> (1.204 s)
--------------------------------------	---	--

- **10 Logger**

Backend Time <b>9 ms</b> (10 ms)	Frontend Time <b>849 ms</b> (881 ms)	Page Load Time <b>858 ms</b> (892 ms)
-------------------------------------	---	--

- **100 Logger**

Backend Time <b>10 ms</b> (18 ms)	Frontend Time <b>1.031 s</b> (1.123 s)	Page Load Time <b>1.043 s</b> (1.143 s)
--------------------------------------	---	--

- **1000 Logger**

Backend Time <b>10 ms</b> (13 ms)	Frontend Time <b>2.031 s</b> (2.209 s)	Page Load Time <b>2.039 s</b> (2.220 s)
--------------------------------------	---	--

## Conclusions and opinions about Webjive benchmark:

The following functions are the way that Webjive handles a new frame of information from the backEnd, all this functions are single thread and code blocking, which means that everytime a new frame of information arrives, Webjive does all this work and just after that it handles a new frame:

**handleNewFrame() -> recordAttribute() -> recordHistory() -> setNewValues() -> setState() -> render()**

### FACTS:

At this point without any change on the code, Webjive can update the simplest widget (attribute display) every ~2ms, which give us around 500 FPS (Frames per second). Still there is one little problem with this, although Webjive is capable of providing this new data every 2ms this is not fully showing on the dashboard for the user, this is, there is a little bug or feature that is missing from really refreshing the visualization, this is captured on ticket AT2-266, for future analysis. (If the user keeps clicking the dashboard the visualization updates otherwise it updates only on idle times)

The widgets that require external libraries (Spectrum, Plot, Scatter, Dial) are not optimized and are not suitable for scaling.

At this point Webjive is working with a single thread (Standard for JavaScript code)

Following the table on the ticket and focusing only on **widgets without external libraries**:

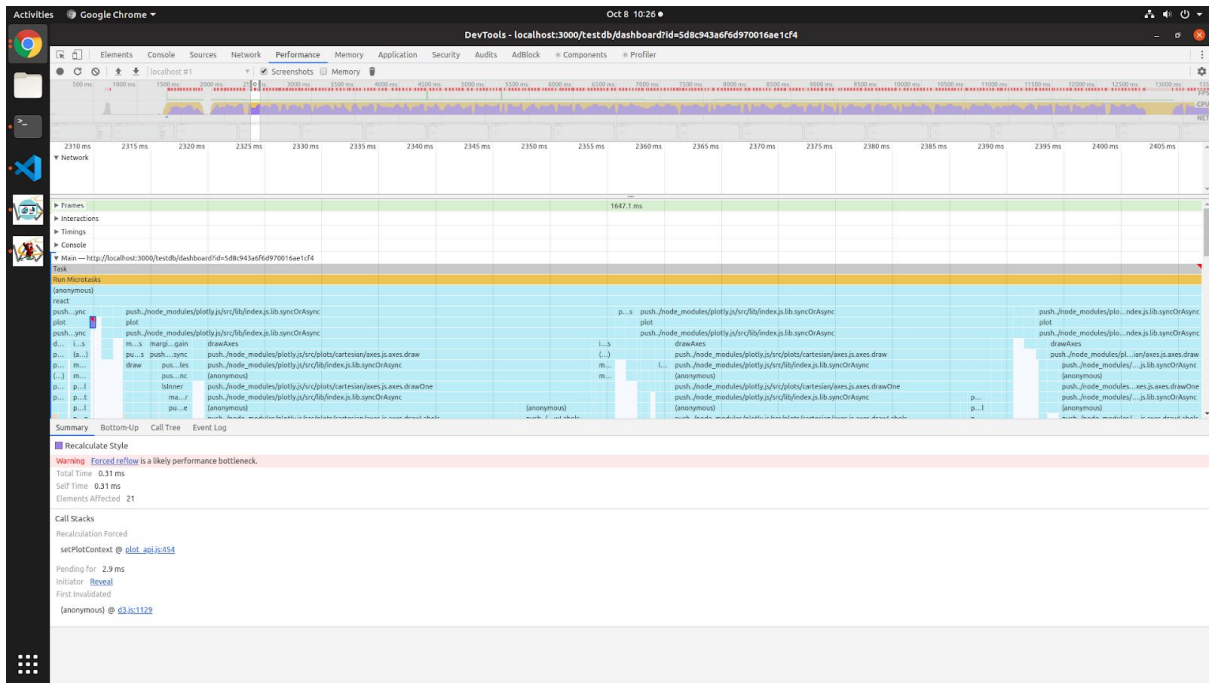
Element	Updates seconds	Can Webjive achieve this as it stands:
1000	1	<b>YES</b>
200	5	<b>YES</b>
1	1000	<b>NO</b>

Another problem gather from the benchmarking was the use of something called **forced reflow**

<https://developers.google.com/web/fundamentals/performance/rendering/avoid-large-complex-layouts-and-layout-thrashing#avoid-forced-synchronous-layouts>

This is a problem related to resizing of widgets that delays the time of displaying/rendering, and it's described as a performance bottleneck





## OPINIONS:

After all the benchmarking done, and some testes, is clear to me that Webjive has many points of possible improvement (manly on JavaScript code), a good start will be to revise all the widgets with external libraries, and maybe create new ones, without external libraries, for sure with less features, but with only the necessary ones for SKA and MAXIV.

Updating Webjive for multithreading (JavaScript workers) code will be fundamental to give it the capability of scaling.

**In my opinion**, if it's SKA desire to have the 1ms update to 1 widget, this technology (all Web browser code) is not suitable for that. (If it's a must-have feature we could migrate to a new C++ native app (optimized also for TangoControls original C++), maybe with QT as UI framework, this will give us full capable multithreading, still a pleasant UI and better performance). We must keep in mind that the companies that make browsers are facing problems with their users, because browsers are taking too much RAM, and there is an ongoing effort to reduce this RAM consumption for every company. (In my mind this is what is causing the AT2-266 problem, not sure, more study about this is required, out of scoop at this point).

If this 1ms / 1000 FPS refresh is not a deal-breaker, it's my opinion that with code optimizations, multithreading in place and dashboards similar to the ones on [Webjive roadmap document](#), then Webjive FRONT END will be able to perform in the way that it's desired by SKA