

---

REPORT:

TermAppISONFT: Orkhestra Cross Test Performance  
Summary

---

Code Magus Limited (England reg. no. 4024745)  
Number 6, 69 Woodstock Road  
Oxford, OX2 6EY, United Kingdom  
[www.codemagus.com](http://www.codemagus.com)  
Copyright © 2021 by Code Magus Limited  
All rights reserved

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Summary of successful outcomes for latest testing</b>	<b>3</b>
2.1	Test 1 - TermAppISONFT - TermAppISO . . . . .	3
<b>3</b>	<b>Comparison of latest tests to pooled previous tests</b>	<b>4</b>
3.1	Differences in response time distributions . . . . .	4
3.1.1	Test 1 - TermAppISONFT - TermAppISO . . . . .	4
3.2	Increases in the response times . . . . .	5
3.3	Decreases in the response times . . . . .	5
3.3.1	Test 1 - TermAppISONFT - TermAppISO . . . . .	5
<b>4</b>	<b>Comparison across all tests individually</b>	<b>6</b>
4.1	Performance of authorisation_request_1100 with outcome: AUTHORISA- TION_RESPONSE_1110_OK . . . . .	7
4.2	Performance of authorisation_request_1100 with outcome: disconnect . . . . .	8
4.3	Performance of authorisation_request_1100 with outcome: timeout . . . . .	9
4.4	Performance of transaction_advice_response_1230 with outcome: disconnect . . . . .	10
4.5	Performance of transaction_advice_response_1230 with outcome: timeout . . . . .	11
4.6	Performance of transaction_advice_response_1230 with outcome: TRANSAC- TION_ADVICE_RESPONSE_1230_OK . . . . .	13
<b>5</b>	<b>Session details</b>	<b>14</b>

```
## Loading required package: lattice
##
## Attaching package: 'BSDA'
## The following object is masked from 'package:datasets':
##
##   Orange
##   : starts: Mon Aug  2 16:43:42 2021
## The following files in ../Test_Summary_Comparisons/csv match pattern "*Performance"
##   File = Test_Performance_Summary_D20201207.csv
##   File = Test_Performance_Summary_D20201208.csv
##   File = Test_Performance_Summary_D20201209_2.csv
##   File = Test_Performance_Summary_D20201209.csv
##   File = Test_Performance_Summary_D20201216.csv
##   File = Test_Performance_Summary_D20210115.csv
##   File = Test_Performance_Summary_D20210510.csv
##   File = Test_Performance_Summary_D20210520.csv
##   File = Test_Performance_Summary_D20210628.csv
##   File = Test_Performance_Summary_D20210629.csv
##   File = Test_Performance_Summary_D20210716.csv
##   File = Test_Performance_Summary_D20210719.csv
##   File = Test_Performance_Summary_D20210722.csv
##   File = Test_Performance_Summary_D20210723.csv
##   File = Test_Performance_Summary_D20210727.csv
##   File = Test_Performance_Summary_D20210728.csv
##   File = Test_Performance_Summary_D20210802.csv
##   File = Test_Performance_Summary_D20201207.csv with 3 rows added to total make
##   File = Test_Performance_Summary_D20201208.csv with 3 rows added to total make
##   File = Test_Performance_Summary_D20201209_2.csv with 5 rows added to total make
##   File = Test_Performance_Summary_D20201209.csv with 3 rows added to total make
##   File = Test_Performance_Summary_D20201216.csv with 15 rows added to total make
##   File = Test_Performance_Summary_D20210115.csv with 3 rows added to total make
##   File = Test_Performance_Summary_D20210510.csv with 4 rows added to total make
##   File = Test_Performance_Summary_D20210520.csv with 4 rows added to total make
##   File = Test_Performance_Summary_D20210628.csv with 5 rows added to total make
##   File = Test_Performance_Summary_D20210629.csv with 3 rows added to total make
##   File = Test_Performance_Summary_D20210716.csv with 3 rows added to total make
##   File = Test_Performance_Summary_D20210719.csv with 3 rows added to total make
##   File = Test_Performance_Summary_D20210722.csv with 5 rows added to total make
##   File = Test_Performance_Summary_D20210723.csv with 6 rows added to total make
##   File = Test_Performance_Summary_D20210727.csv with 3 rows added to total make
##   File = Test_Performance_Summary_D20210728.csv with 3 rows added to total make
##   File = Test_Performance_Summary_D20210802.csv with 3 rows added to total make
```

## 1 Introduction

There are three elements to this cross-test performance summary report. The first shows a summary of the percentage success of each function/operation/call. The second element compares the performance of the latest test(s) to the pooled performance of previous tests for each of the functions/operations/calls and outcomes. And the third element of this report compares the performance by function/operation/call by the outcome across multiple NFT result sets.

The percentage successful outcomes are presented as a summary for the latest test(s). This summary is ranked in increasing order of the percentage of good outcomes of that scenario against all attempts of that scenario in the test.

As a summary and for ranking the performance of the last tests results against previous test results, for each function/operation/call and outcome, the tests in the last test session are compared against the tests in previous sessions. This is accomplished by pooling the sample mean of the response times and pooling the sample standard deviations of response times across all prior tests, and then comparing the latest test(s) with the previous tests using `tsum.test`. The results are by ranked by the corresponding  $p$ -values in increasing order and tabulated. For each function/operation/call request, three comparison tests are made: The first determines a measure of the difference between the respective response time distributions; the second determines a measure of those response times that could be considered worse in the latest test(s) as compared to the pooled previous test; and the third determines a measure of those response times that could be considered better in the latest test(s) as compared to the pooled previous tests.

In addition to tabulating the response time means and standard deviations against function/operation/call and outcomes across the tests, box-plots are produced to visually compare the performance/outcomes over the various tests. In each case, the box-plots show up the 15 most extreme functions/operations/calls that are most different to the historic response time distributions, and then a box-plot each showing those that have response times greatest increase and decrease in their response times when compared to their respective historic counterparts.

The last section of the report compares the performance by function/operation/call by the outcome across multiple NFT result sets. The summary results have been taken from the application performance sections of the individual NFT sessions. The `Resp` value is the sample mean of the response times in seconds and the `StdDev` the corresponding sample standard deviation. In each case only those values where the customer or business function arrival rate did not materially exceed the peak observed/production target are included in the calculation.

## 2 Summary of successful outcomes for latest testing

### 2.1 Test 1 - TermAppISONFT - TermAppISO

The following table is a summary of the outcomes of test 1 (TermAppISONFT - TermAppISO), showing the percentage of functions/operations/calls considered successful. The scenarios are shown from worst percentage good outcomes to best:

StartTime	TestNum	Label	Description	Basename	Outcome	Count	PercentResp	StdDev
2021-08-02 13:24:01	1	TermAppISONFT	Transaction_advice_response	TRANSACTION_ADVICE_RESPONSE	OK	102306	100	0.060
2021-08-02 13:24:01	1	TermAppISONFT	Authorisation_request	AUTHORISATION_REQUEST	OK	100020	100	0.061

### 3 Comparison of latest tests to pooled previous tests

The last test date in the summary data is used to delimit the prior tests from the tests in the last test session. This section compares the tests performed on `testdate` to the tests that ran in sessions prior to this date. Comparisons are made only for the successful outcomes, and only the performance data where the rate in each of the tests included in the comparison did not exceed the target rate is included in the comparison.

#### 3.1 Differences in response time distributions

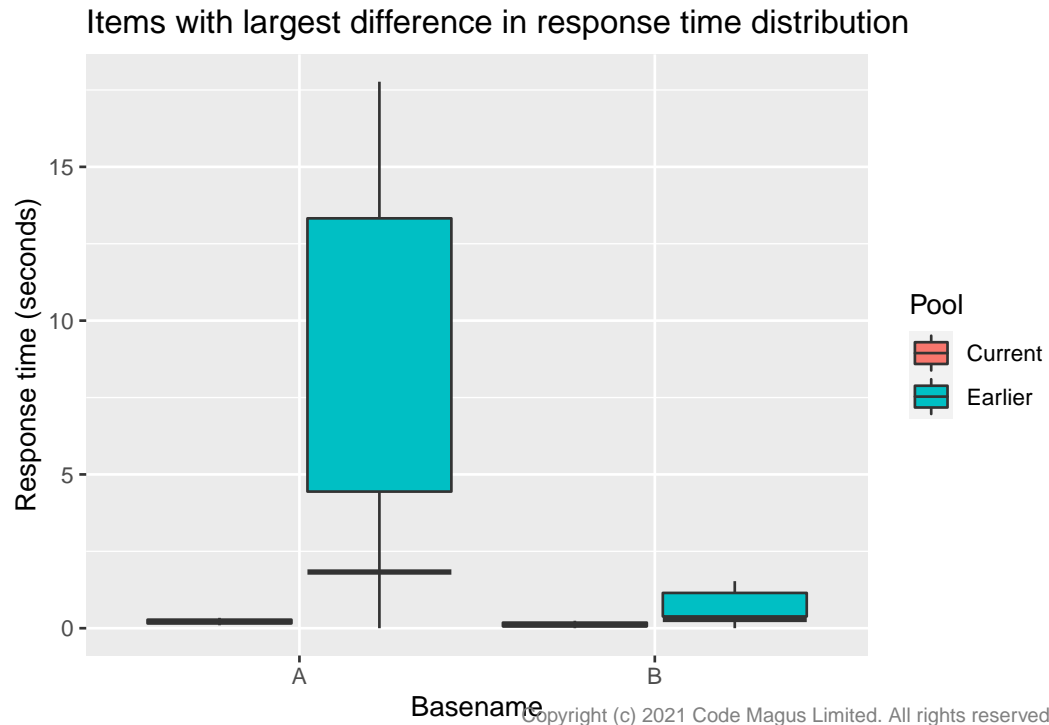
The following show the comparisons of the good outcomes of the tests performed on 2021-08-02 as compared to the tests performed before this date. The table is ranked in increasing order of the  $p$ -values from the corresponding Welch Modified Two-Sample t-Test (two.sided), starting from the function/operation/call where the response time distribution differences are the greatest. Results are only shown for which the  $p$ -value is less than or equal to the cutoff value ( $\alpha = 0.05$ ).

##### 3.1.1 Test 1 - TermAppISONFT - TermAppISO

The following compare the responses time differences from the test started at 2021-08-02 13:24:01 to the tests from previous test sessions.

Basename	Outcome	Count	Resp	StdDev	PrevCount	PrevMean	PrevStdDev	pvalue.d
authorisation_request_1100	AUTHORISATION_RESPONSE	102306	100	0.061	251757	1.827	4.442	0
transaction_advice_response_1200	TRANSACTION_ADVICE_RESPONSE	100020	100	0.060	50385	0.285	0.382	0

```
## Loading required package: grid
```



Key	Basename
A	authorisation_request_1100
B	transaction_advice_response_1230

### 3.2 Increases in the response times

There were no significant response time increases when comparing the test(s) in the last test session to tests from earlier test sessions for any of the items.

### 3.3 Decreases in the response times

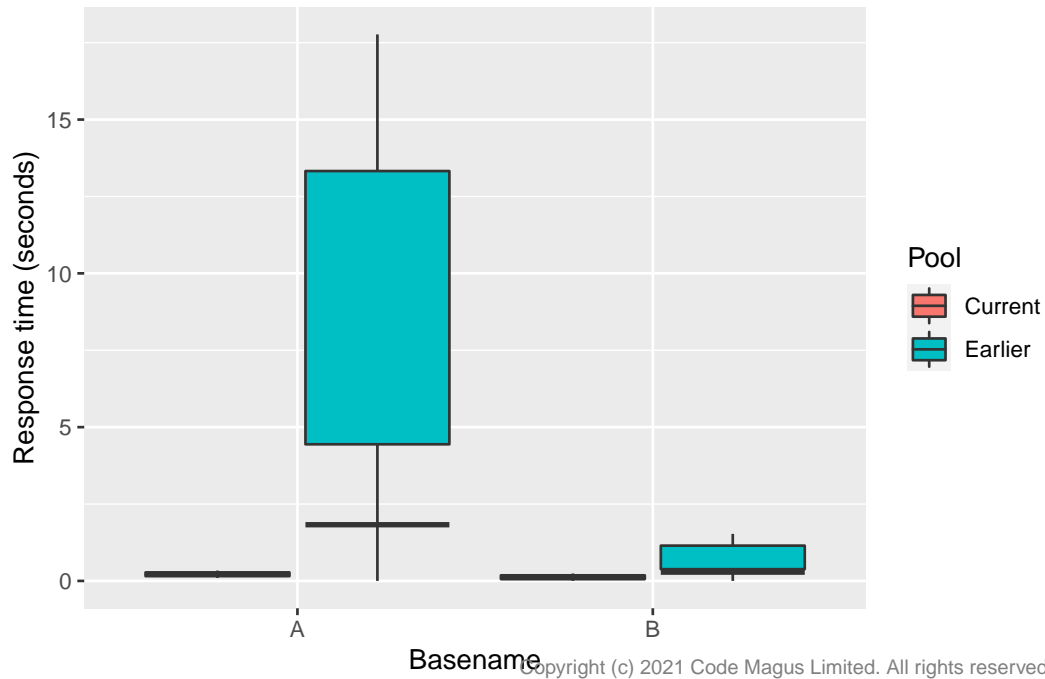
The following show the comparisons of the good outcomes of the tests performed on 2021-08-02 as compared to the tests performed before this date. The table is ranked in increasing order of the  $p$ -values from the corresponding Welch Modified Two-Sample t-Test (less), starting from the function/operation/call where the response time decreases are the greatest. Results are only shown for which the  $p$ -value is less than or equal to the cutoff value ( $\alpha = 0.05$ ).

#### 3.3.1 Test 1 - TermAppISONFT - TermAppISO

The following compare the responses time decreases from the test started at 2021-08-02 13:24:01 to the tests from previous test sessions.

Basename	Outcome	Count	Resp	StdDev	PrevCount	PrevMean	PrevStdDev	pvalue
authorisation_request_1100	AUTHORISATION_RESPONSE	508	10.206	0.061	251757	1.827	4.442	0
transaction_advice_response_1230	TRANSACTION_ADVICE_RESPONSE	1201	0.285	0.001	250385	0.285	0.382	0

Items with largest decrease in response times



Key	Basename
A	authorisation_request_1100
B	transaction_advice_response_1230

## 4 Comparison across all tests individually

This section compares the performance between the NFT tests to date for each of the functions/operations/calls included in the corresponding test.

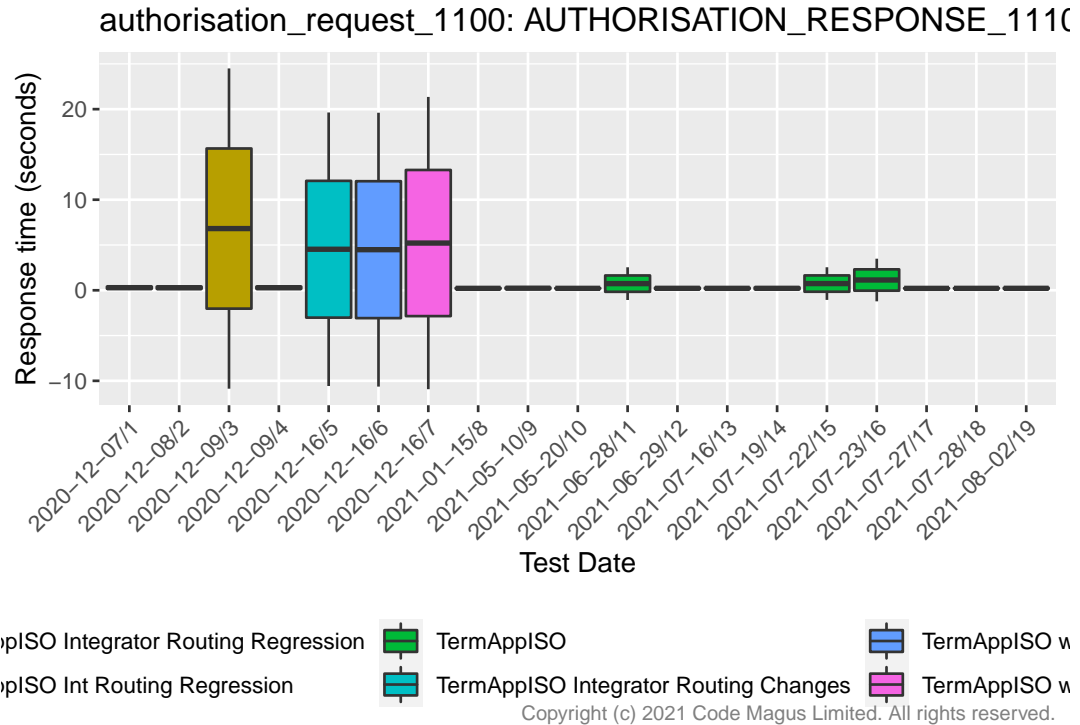
In the box-plots that follow, in each case, the centre is the sample mean response time value in seconds for that particular function/operation/call qualified by the outcome of that function/operation/call. The lower edge of the box is the corresponding sample mean response time value less the sample standard deviation, and the upper edge of the box is the corresponding sample mean response time value plus the standard deviation. The minimum and maximum values are calculated by taking two times the standard deviation in a similar manner.

#### 4.1 Performance of authorisation\_request\_1100 with outcome: AUTHORIZATION\_RESPONSE\_1110\_OK

The following table shows the performance descriptive statistics for authorisation\_request\_1100 when the outcomes are AUTHORIZATION\_RESPONSE\_1110\_OK.

TestDate	Description	Basename	Outcome	Count	Percent	Resp	StdDev
2020-12-07	TermAppISO Integrator Routing Regression	authorisation_request	AUTHORIZATION_RESPONSE_1110_OK	560	100.0000	0.087	0.123
2020-12-08	TermAppISO Integrator Routing Regression	authorisation_request	AUTHORIZATION_RESPONSE_1110_OK	560	100.0000	0.081	0.124
2020-12-09	TermAppISO Int Routing Regression	authorisation_request	AUTHORIZATION_RESPONSE_1110_OK	18216	100.0000	0.083	8.841
2020-12-09	TermAppISO	authorisation_request	AUTHORIZATION_RESPONSE_1110_OK	560	100.0000	0.083	0.122
2020-12-16	TermAppISO Integrator Routing Changes	authorisation_request	AUTHORIZATION_RESPONSE_1110_OK	9541	100.0000	0.081	7.550
2020-12-16	TermAppISO with 40 provider threads	authorisation_request	AUTHORIZATION_RESPONSE_1110_OK	9741	100.0000	0.083	7.560
2020-12-16	TermAppISO with 80 provider threads	authorisation_request	AUTHORIZATION_RESPONSE_1110_OK	9431	100.0000	0.084	8.069
2021-01-15	TermAppISO	authorisation_request	AUTHORIZATION_RESPONSE_1110_OK	560	100.0000	0.083	0.049
2021-05-10	TermAppISO	authorisation_request	AUTHORIZATION_RESPONSE_1110_OK	1001	100.0000	0.089	0.074
2021-05-20	TermAppISO	authorisation_request	AUTHORIZATION_RESPONSE_1110_OK	1021	100.0000	0.083	0.028
2021-06-28	TermAppISO	authorisation_request	AUTHORIZATION_RESPONSE_1110_OK	1075	100.0000	0.082	0.903
2021-06-29	TermAppISO	authorisation_request	AUTHORIZATION_RESPONSE_1110_OK	560	100.0000	0.085	0.079
2021-07-16	TermAppISO	authorisation_request	AUTHORIZATION_RESPONSE_1110_OK	560	100.0000	0.083	0.062
2021-07-19	TermAppISO	authorisation_request	AUTHORIZATION_RESPONSE_1110_OK	560	100.0000	0.087	0.100
2021-07-22	TermAppISO	authorisation_request	AUTHORIZATION_RESPONSE_1110_OK	6941	100.0000	0.089	0.901
2021-07-23	TermAppISO	authorisation_request	AUTHORIZATION_RESPONSE_1110_OK	6971	100.0000	0.083	1.177
2021-07-27	TermAppISO	authorisation_request	AUTHORIZATION_RESPONSE_1110_OK	560	100.0000	0.080	0.030
2021-07-28	TermAppISO	authorisation_request	AUTHORIZATION_RESPONSE_1110_OK	560	100.0000	0.082	0.035
2021-08-02	TermAppISO	authorisation_request	AUTHORIZATION_RESPONSE_1110_OK	560	100.0000	0.086	0.061



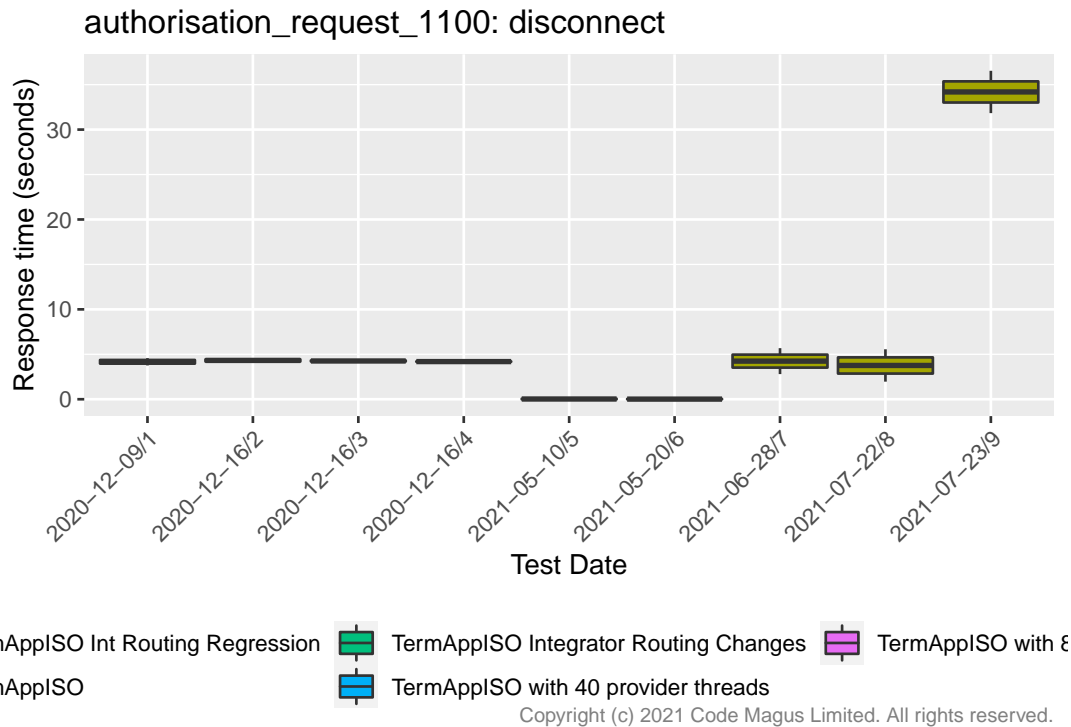


## 4.2 Performance of authorisation\_request\_1100 with outcome: disconnect

The following table shows the performance descriptive statistics for authorisation\_request\_1100 when the outcomes are disconnect.

TestDate	Description	Basename	Outcome	Count	Percent	Resp	StdDev
2020-12-09	TermAppISO Int Routing Regression	authorisation_request_1100	disconnect	304	1.602	4.166	0.205
2020-12-16	TermAppISO Integrator Routing Changes	authorisation_request_1100	disconnect	3	0.015	4.318	0.124
2020-12-16	TermAppISO with 40 provider threads	authorisation_request_1100	disconnect	2	0.010	4.258	0.060
2020-12-16	TermAppISO with 80 provider threads	authorisation_request_1100	disconnect	5	0.026	4.191	0.021
2021-05-10	TermAppISO	authorisation_request_1100	disconnect	7173	50.000	0.027	0.037
2021-05-20	TermAppISO	authorisation_request_1100	disconnect	7089	49.979	0.016	0.024
2021-06-28	TermAppISO	authorisation_request_1100	disconnect	467	2.142	4.241	0.721
2021-07-22	TermAppISO	authorisation_request_1100	disconnect	568	2.579	3.758	0.901
2021-07-23	TermAppISO	authorisation_request_1100	disconnect	177	1.257	34.187	1.177

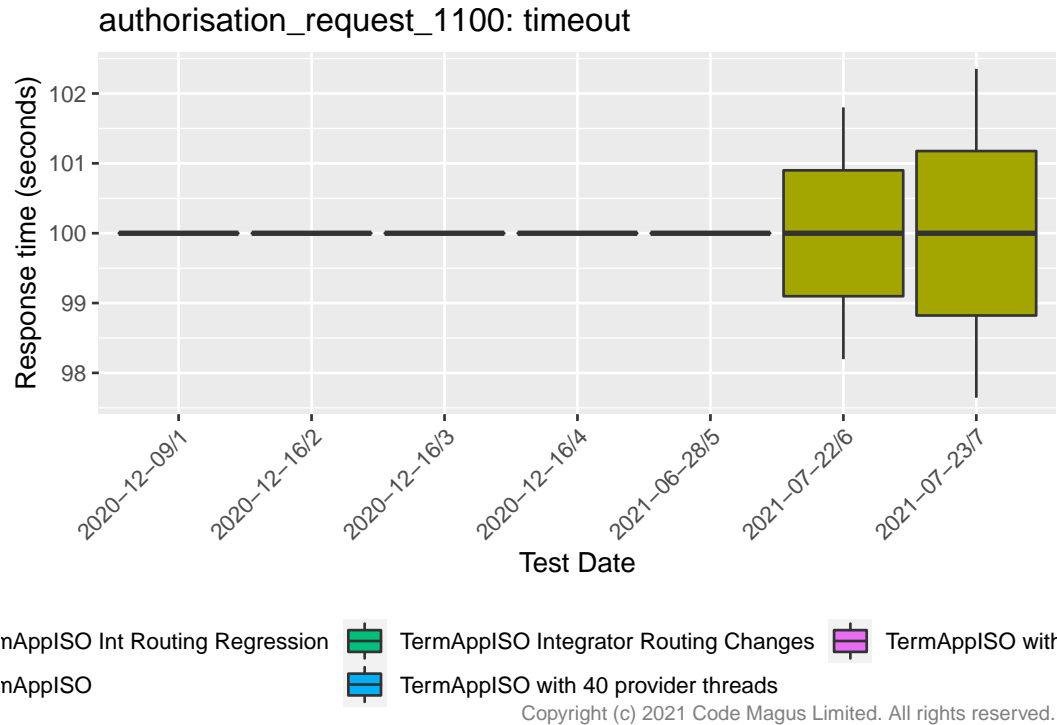
TestDate	Description	Basename	Outcome	Count	Percent	Resp	StdDev
----------	-------------	----------	---------	-------	---------	------	--------



### 4.3 Performance of authorisation\_request\_1100 with outcome: timeout

The following table shows the performance descriptive statistics for authorisation\_request\_1100 when the outcomes are timeout.

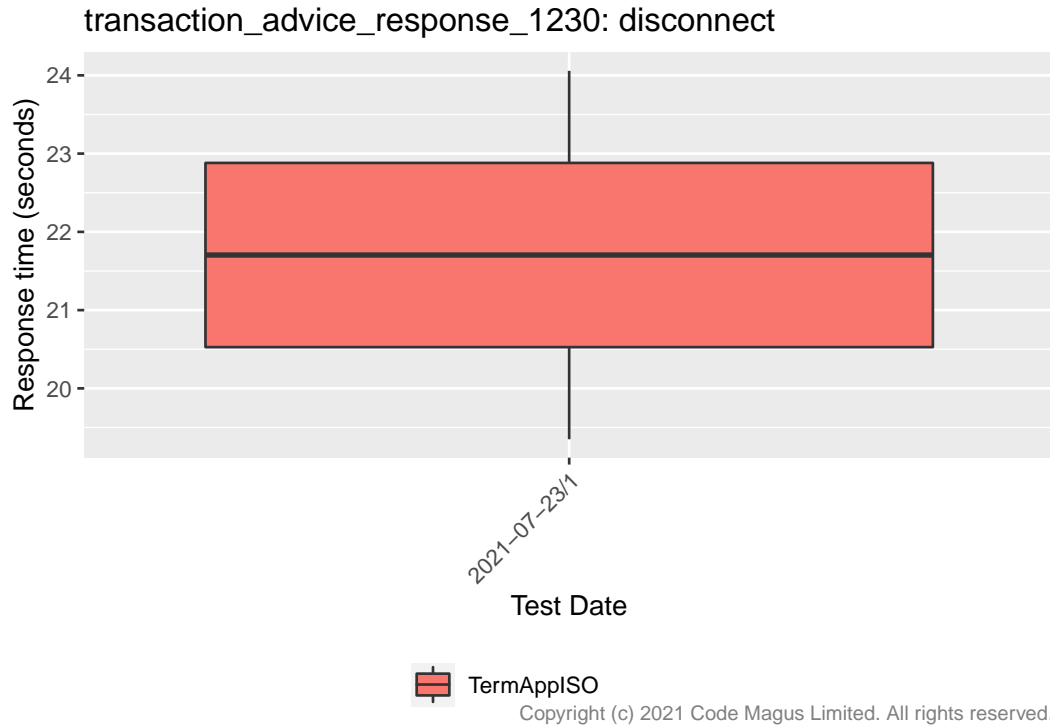
TestDate	Description	Basename	Outcome	Count	Percent	Resp	StdDev
2020-12-09	TermAppISO Int Routing Regression	authorisation_request_1100	timeout	41	0.216	99.999	0.000
2020-12-16	TermAppISO Integrator Routing Changes	authorisation_request_1100	timeout	6	0.031	99.999	0.000
2020-12-16	TermAppISO with 40 provider threads	authorisation_request_1100	timeout	3	0.015	99.999	0.000
2020-12-16	TermAppISO with 80 provider threads	authorisation_request_1100	timeout	6	0.031	99.999	0.000
2021-06-28	TermAppISO	authorisation_request_1100	timeout	17396	79.784	99.999	0.000
2021-07-22	TermAppISO	authorisation_request_1100	timeout	17342	78.727	99.999	0.901
2021-07-23	TermAppISO	authorisation_request_1100	timeout	1696	12.045	99.999	1.177



#### 4.4 Performance of transaction\_advice\_response\_1230 with outcome: disconnect

The following table shows the performance descriptive statistics for transaction\_advice\_response\_1230 when the outcomes are disconnect.

TestDate	Description	Basename	Outcome	Count	Percent	Resp	StdDev
2021-07-23	TermAppISO	transaction_advice_response_1230	disconnect	2	0.016	21.704	1.177

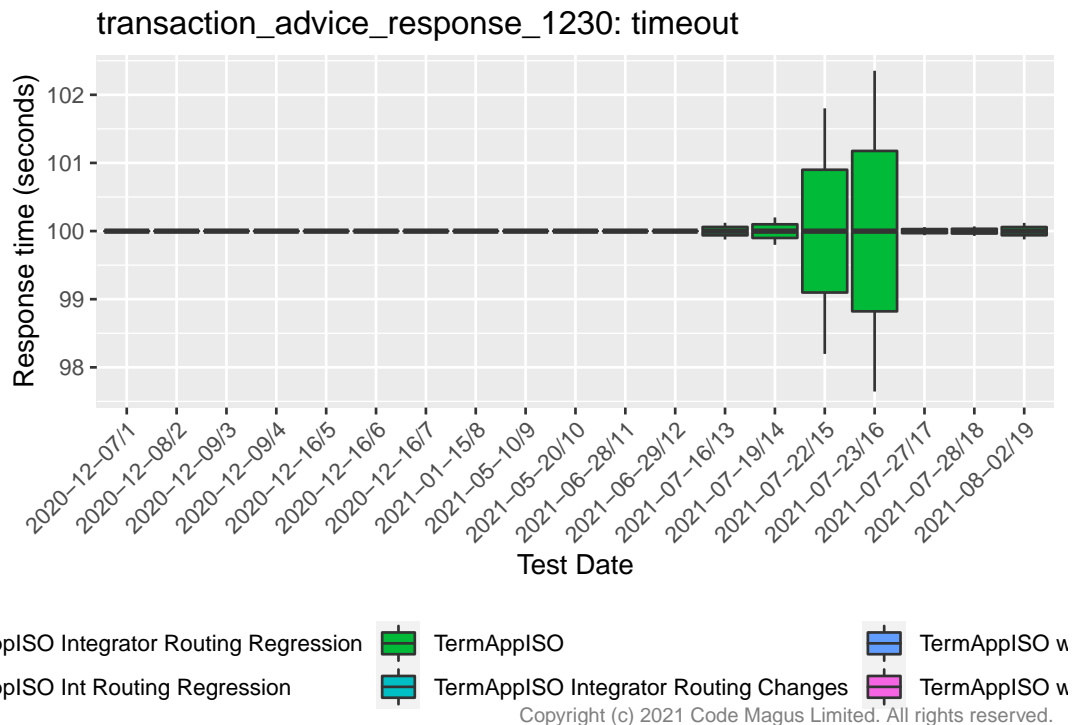


#### 4.5 Performance of transaction\_advice\_response\_1230 with outcome: timeout

The following table shows the performance descriptive statistics for transaction\_advice\_response\_1230 when the outcomes are timeout.

TestDate	Description	Basename	OutcomeCount	Percent	Resp	StdDev
2020-12-07	TermAppISO Integrator Routing Regression	transaction_advice_response_timeout	33	0.211	99.999	0.000
2020-12-08	TermAppISO Integrator Routing Regression	transaction_advice_response_timeout	16	0.101	99.999	0.000
2020-12-09	TermAppISO Int Routing Regression	transaction_advice_response_timeout	15	0.081	99.999	0.000
2020-12-09	TermAppISO	transaction_advice_response_timeout	35	0.214	99.999	0.000
2020-12-16	TermAppISO Integrator Routing Changes	transaction_advice_response_timeout	28	0.145	99.999	0.000
2020-12-16	TermAppISO with 40 provider threads	transaction_advice_response_timeout	16	0.082	99.999	0.000
2020-12-16	TermAppISO with 80 provider threads	transaction_advice_response_timeout	57	0.296	99.999	0.000
2021-01-15	TermAppISO	transaction_advice_response_timeout	20	0.122	99.999	0.000

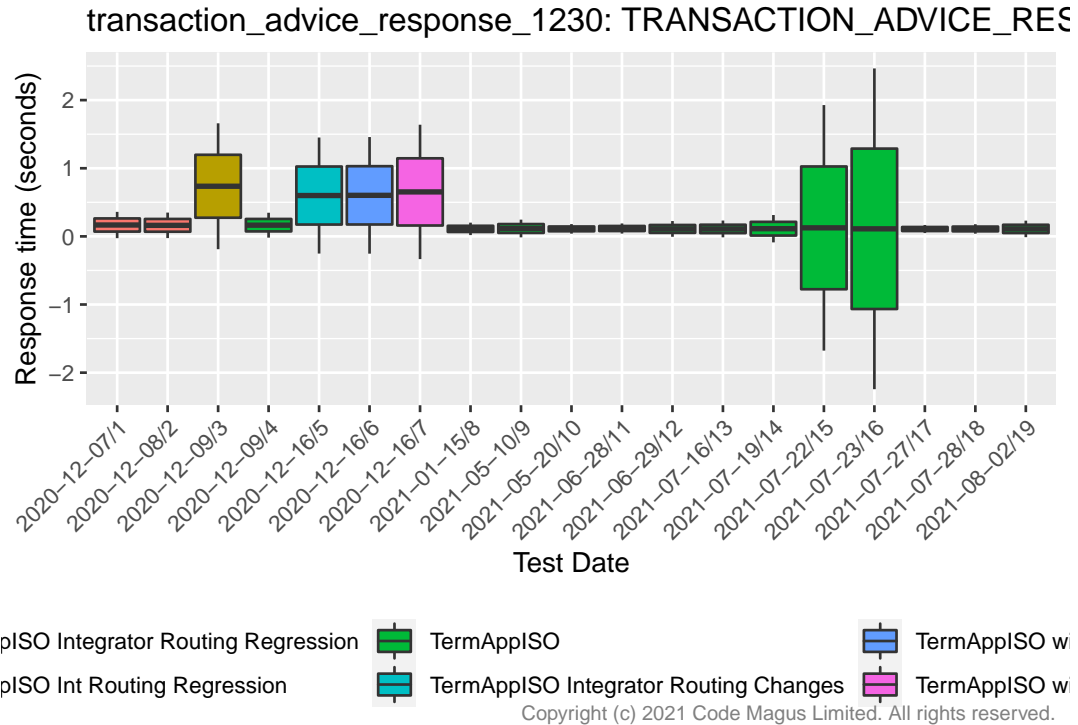
TestDate	Description	Basename	Outcome	Count	Percent	Resp	StdDev
2021-05-10	TermAppISO	transaction_advice_response_timeout	fail	14	0.196	99.999	0.000
2021-05-20	TermAppISO	transaction_advice_response_timeout	fail	13	0.184	99.999	0.000
2021-06-28	TermAppISO	transaction_advice_response_timeout	fail	14	0.357	99.999	0.000
2021-06-29	TermAppISO	transaction_advice_response_timeout	fail	17	0.122	99.999	0.000
2021-07-16	TermAppISO	transaction_advice_response_timeout	fail	23	0.156	99.999	0.062
2021-07-19	TermAppISO	transaction_advice_response_timeout	fail	32	0.205	99.999	0.100
2021-07-22	TermAppISO	transaction_advice_response_timeout	fail	18	0.439	99.999	0.901
2021-07-23	TermAppISO	transaction_advice_response_timeout	fail	17	0.140	99.999	1.177
2021-07-27	TermAppISO	transaction_advice_response_timeout	fail	16	0.103	99.999	0.030
2021-07-28	TermAppISO	transaction_advice_response_timeout	fail	15	0.095	99.999	0.035
2021-08-02	TermAppISO	transaction_advice_response_timeout	fail	28	0.191	99.999	0.061



#### 4.6 Performance of transaction\_advice\_response\_1230 with outcome: TRANSACTION\_ADVICE\_RESPONSE\_1230\_OK

The following table shows the performance descriptive statistics for transaction\_advice\_response\_1230 when the outcomes are TRANSACTION\_ADVICE\_RESPONSE\_1230\_OK.

TestDate	Description	Basename	Outcome	Count	PercentResp	StdDev
2020-12-07	TermAppISO Integrator Routing Regression	transaction_advice_res	TRANSACTION_ADVICE_RESPONSE_1230_OK	1570	100%	1623000K
2020-12-08	TermAppISO Integrator Routing Regression	transaction_advice_res	TRANSACTION_ADVICE_RESPONSE_1230_OK	1780	100%	1623000K
2020-12-09	TermAppISO Int Routing Regression	transaction_advice_res	TRANSACTION_ADVICE_RESPONSE_1230_OK	1538	100%	17363046K
2020-12-09	TermAppISO	transaction_advice_res	TRANSACTION_ADVICE_RESPONSE_1230_OK	1638	100%	16423000K
2020-12-16	TermAppISO Integrator Routing Changes	transaction_advice_res	TRANSACTION_ADVICE_RESPONSE_1230_OK	1290	100%	1561023042K
2020-12-16	TermAppISO with 40 provider threads	transaction_advice_res	TRANSACTION_ADVICE_RESPONSE_1230_OK	1503	100%	1561023042K
2020-12-16	TermAppISO with 80 provider threads	transaction_advice_res	TRANSACTION_ADVICE_RESPONSE_1230_OK	1278	100%	156123049K
2021-01-15	TermAppISO	transaction_advice_res	TRANSACTION_ADVICE_RESPONSE_1230_OK	1635	100%	1611023004K
2021-05-10	TermAppISO	transaction_advice_res	TRANSACTION_ADVICE_RESPONSE_1230_OK	1389	100%	1611023006K
2021-05-20	TermAppISO	transaction_advice_res	TRANSACTION_ADVICE_RESPONSE_1230_OK	105	100%	1611023006K
2021-06-28	TermAppISO	transaction_advice_res	TRANSACTION_ADVICE_RESPONSE_1230_OK	191	100%	1611523006K
2021-06-29	TermAppISO	transaction_advice_res	TRANSACTION_ADVICE_RESPONSE_1230_OK	187	100%	1611023006K
2021-07-16	TermAppISO	transaction_advice_res	TRANSACTION_ADVICE_RESPONSE_1230_OK	173	100%	1610923006K
2021-07-19	TermAppISO	transaction_advice_res	TRANSACTION_ADVICE_RESPONSE_1230_OK	153	100%	161123010K
2021-07-22	TermAppISO	transaction_advice_res	TRANSACTION_ADVICE_RESPONSE_1230_OK	108	100%	1612523090K
2021-07-23	TermAppISO	transaction_advice_res	TRANSACTION_ADVICE_RESPONSE_1230_OK	126	100%	161123010K
2021-07-27	TermAppISO	transaction_advice_res	TRANSACTION_ADVICE_RESPONSE_1230_OK	150	100%	1610823006K
2021-07-28	TermAppISO	transaction_advice_res	TRANSACTION_ADVICE_RESPONSE_1230_OK	170	100%	1610923006K
2021-08-02	TermAppISO	transaction_advice_res	TRANSACTION_ADVICE_RESPONSE_1230_OK	146	100%	1611023006K



## 5 Session details

```
sessionInfo();
```

```
## R version 3.6.0 (2019-04-26)
## Platform: x86_64-redhat-linux-gnu (64-bit)
## Running under: CentOS Linux 7 (Core)
##
## Matrix products: default
## BLAS/LAPACK: /usr/lib64/R/lib/libRblas.so
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8      LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8      LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8  LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8     LC_NAME=C
##  [9] LC_ADDRESS=C             LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] grid      stats    graphics grDevices utils    datasets methods
## [8] base
##
```

```
## other attached packages:
## [1] pander_0.6.3      doBy_4.6.7        cmlrutils_1.19   XML_3.98-1.20
## [5] scales_1.1.1      ggplot2_3.3.2    BSDA_1.2.0       lattice_0.20-38
##
## loaded via a namespace (and not attached):
## [1] Rcpp_1.0.5         highr_0.8         pillar_1.4.6     compiler_3.6.0
## [5] class_7.3-15      tools_3.6.0       digest_0.6.25    evaluate_0.14
## [9] lifecycle_0.2.0  tibble_3.0.3     gtable_0.3.0     pkgconfig_2.0.3
## [13] rlang_0.4.7       Matrix_1.2-17    yaml_2.2.1       xfun_0.17
## [17] e1071_1.7-4      withr_2.2.0      stringr_1.4.0    dplyr_1.0.2
## [21] knitr_1.30        generics_0.0.2   vctrs_0.3.2     tidyselect_1.1.0
## [25] glue_1.4.1       R6_2.4.1         rmarkdown_2.6    farver_2.0.3
## [29] tidyr_1.1.2      purrr_0.3.4      cmlbrandr_3.0    magrittr_1.5
## [33] backports_1.1.8  ellipsis_0.3.1  htmltools_0.5.0  MASS_7.3-51.4
## [37] colorspace_1.4-1 Deriv_4.0.1      labeling_0.3     stringi_1.5.3
## [41] munsell_0.5.0    broom_0.7.0      crayon_1.3.4
```