On the Statistical Analysis of Computer Performance Test Data

Presentation
CMG IMPACT 2025
Natick, Massachusetts

Stephen R Donaldson

2025-06-02

Outline

The talk is broken down as follows:

1. Problem statement
2. The field environment
3. The test environment
4. Gathering metric data
5. Checking traits
6. Methods
7. Example
8. Conclusion

Problem Statement

Problem statement

Insufficiency of Traditional Performance Metrics: Traditional externally observed metrics like response time and throughput may not provide enough insight into a system's readiness for deployment. These metrics often lack the context of resource usage and internal system behaviour, leading to potential misinterpretations.

Lack of Formal Experimental Design in Performance Testing: Ad-hoc or poorly structured performance testing may yield unreliable conclusions. The study argues for conducting performance testing as a formal experimental study with controlled design, execution, and analysis to improve the reliability of conclusions.

Need for Augmented and Synchronized Data: Often, necessary data for post-hoc analysis is missing, misaligned, or unsuitable. The study stresses the importance of synchronizing and augmenting test data with resource usage metrics for meaningful analysis.

Understanding Resource Costs and Bottlenecks: There's a need to quantify how system load (e.g., user activity) affects resource usage (CPU, network bandwidth). This helps in identifying bottlenecks, estimating resource cost per user, and detecting inefficiencies or anomalies in system behaviour.

Improving Decision-Making for Field Deployment: The ultimate aim is to better inform decisions about software deployment. This includes determining whether systems meet required performance thresholds under expected loads and understanding if capacity planning is adequate.

Highlighting the Importance of Statistical Tools: The study demonstrates the application of statistical methods—especially linear regression, local regression, and bootstrap techniques—to derive actionable insights from performance data.

Automation and Efficiency in Analysis: Given the volume of data (often tens of millions of observations), the need for automated, reproducible, and efficient data analysis pipelines is critical.

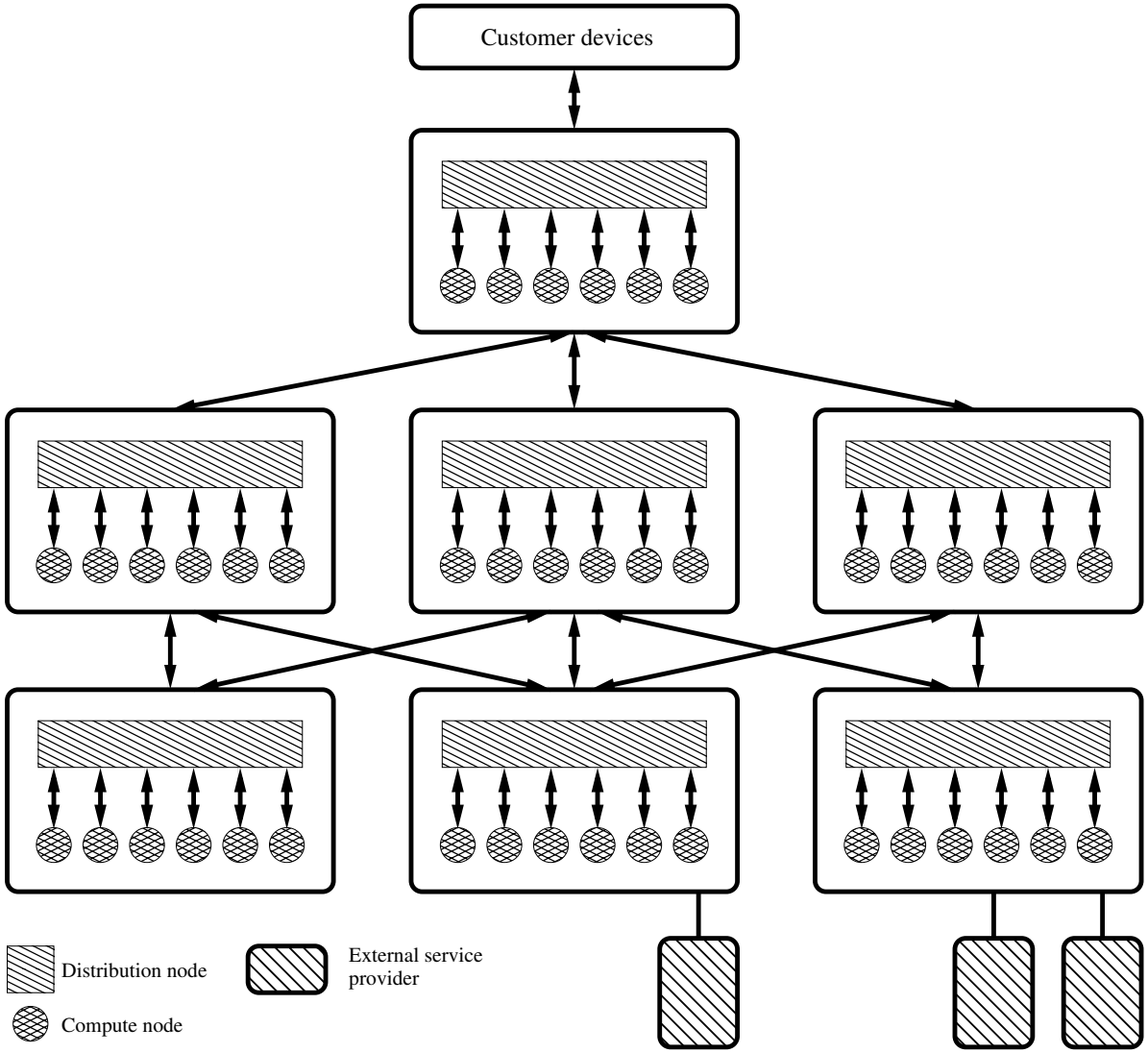
Servers/appliances and their clusters

Individual **servers** (or **appliances**) make up the units that support an application function or some facet of an application's function — for example, a service providing the functionality to deliver access controls for users would typically be viewed as an application function, and distinct from the application function upon which the database management system runs that maintains the state of user's access and credentials.

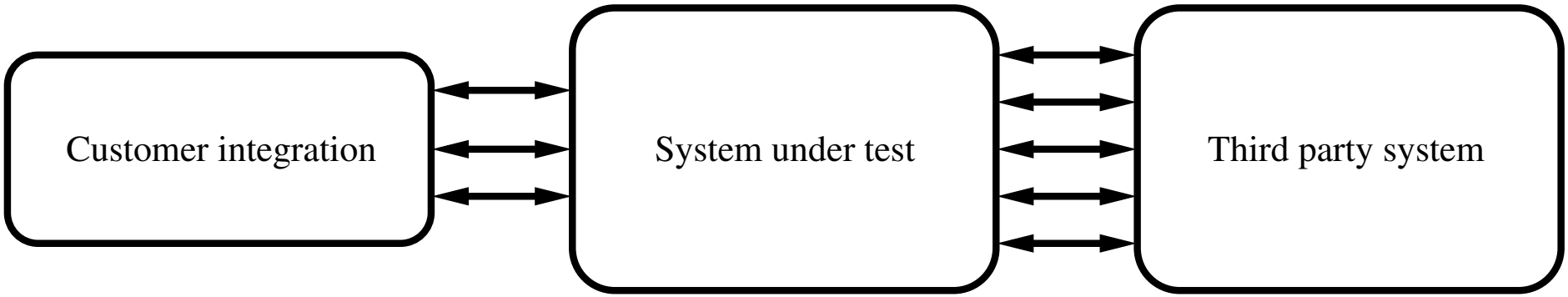
Servers/appliances provisioned to host the same functions are typically arranged as members of a **cluster**. Members of a cluster may operate concurrently in an **Active/Active** arrangement (where all or some of the members concurrently process with the workload) or in an **Active/Passive** arrangement (where servers/appliances are designated Active and processing workload, or Passive and on standby).

Clusters, in which multiple components are expected to independently process the workload, typically include a **load balancing** function responsible for distributing the workload.

Field landscape

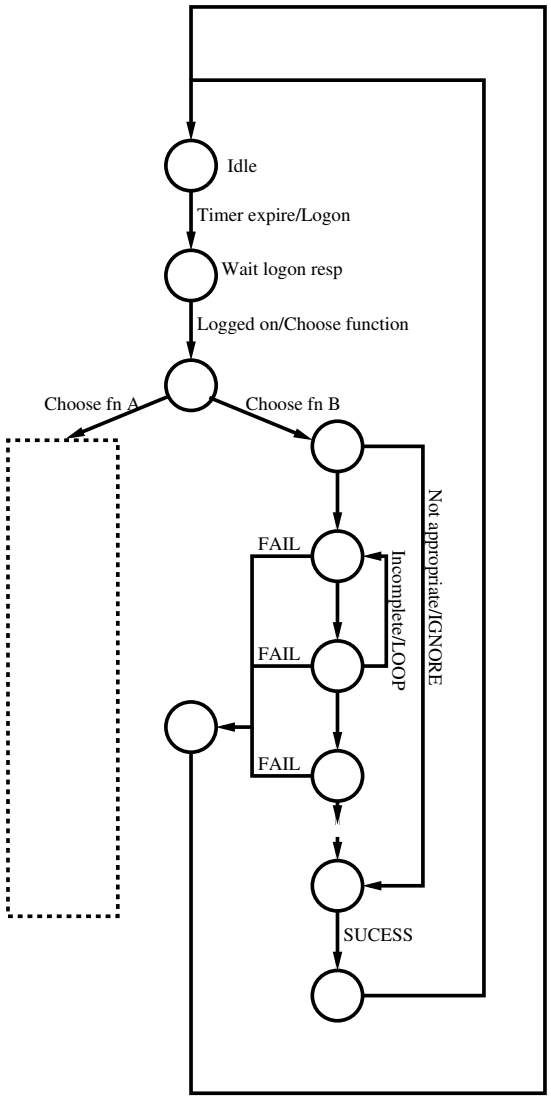


Test environment



Modelling customer interaction

- Markov model - Finite state, continuous time
- Extended FSM or I/O FSM
- Behaviour driven by model parameters and responses from system-under-test (both transition probabilities and sojourn times)
- Model parameters estimated from field metrics, contracts, SME estimation, etc.



Modelling third-party system interactions

- Multiple third-parties
- Hidden features and observable features
- Parameters estimated from field metrics

KDE:

$$\widehat{f}(x) = \frac{1}{nb}\sum_{i=1}^{n} K\left(\frac{x-x_i}{b}\right)$$

Tukey-g-and-h:

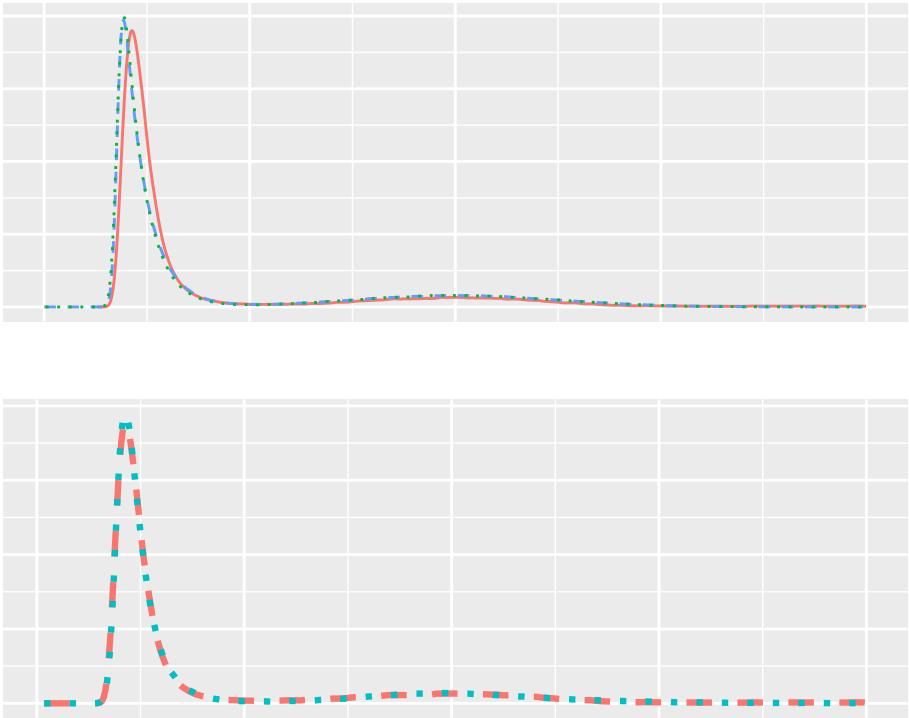
$$T_{g,h} \sim \frac{e^{gZ}-1}{g} e^{hZ^2/2}$$

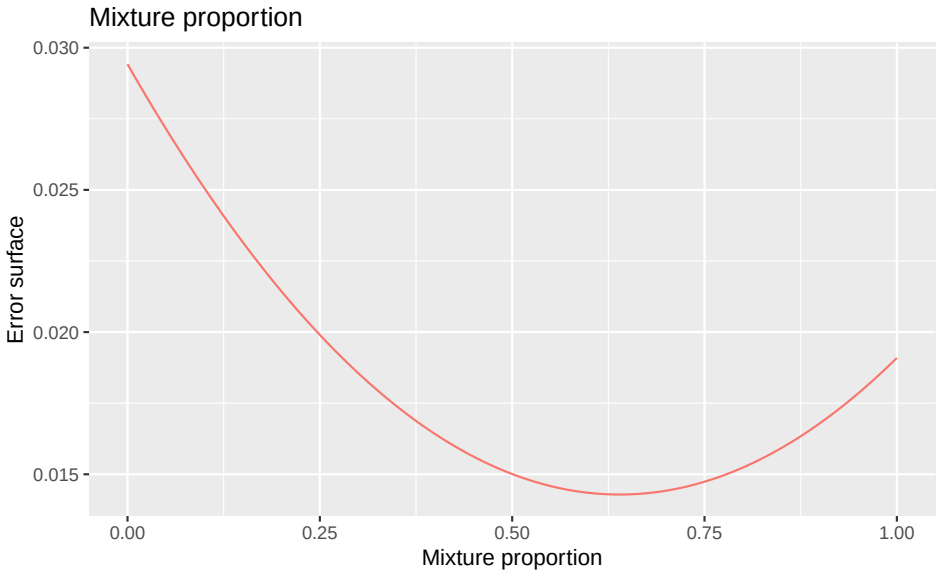
where $Z \sim N(0,1)$

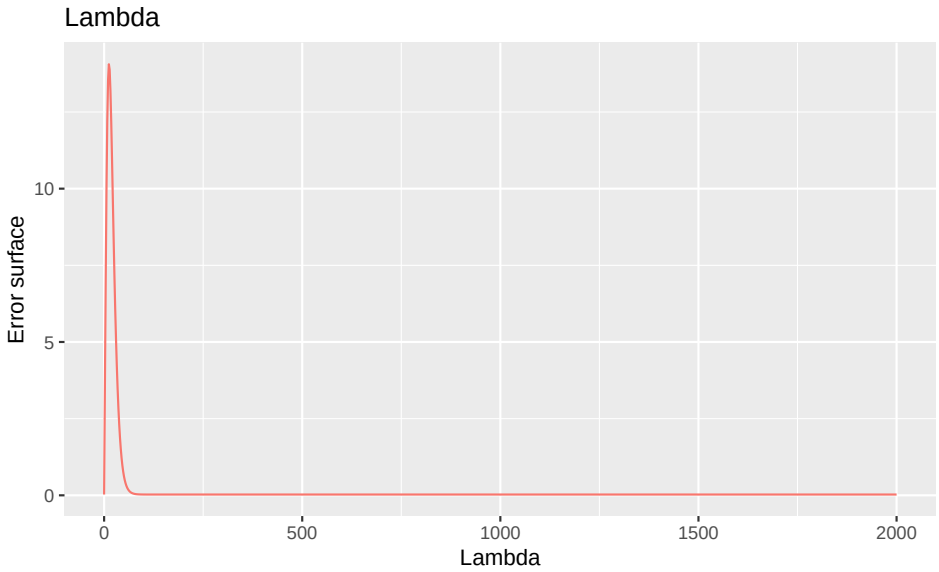
and g skewness, h tail density

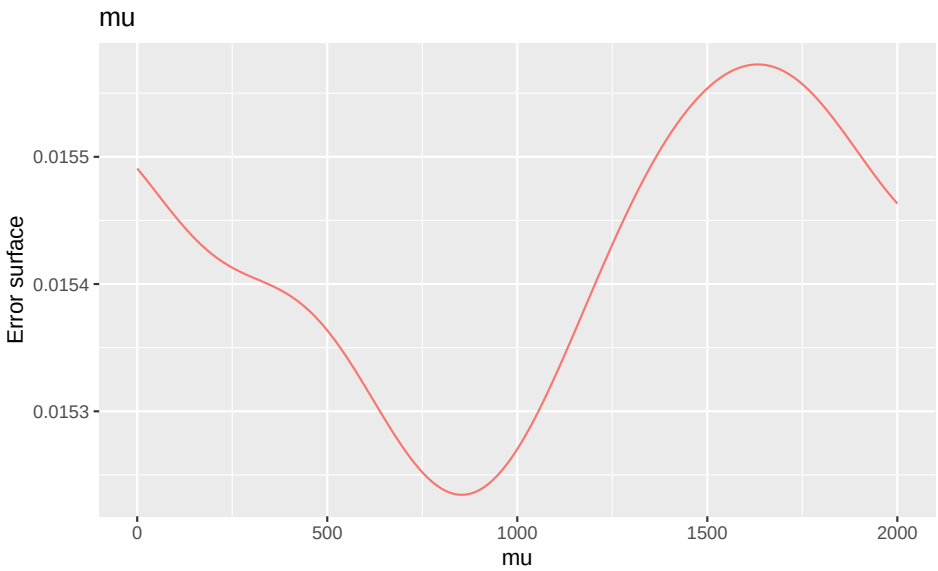
Finite mixtures:

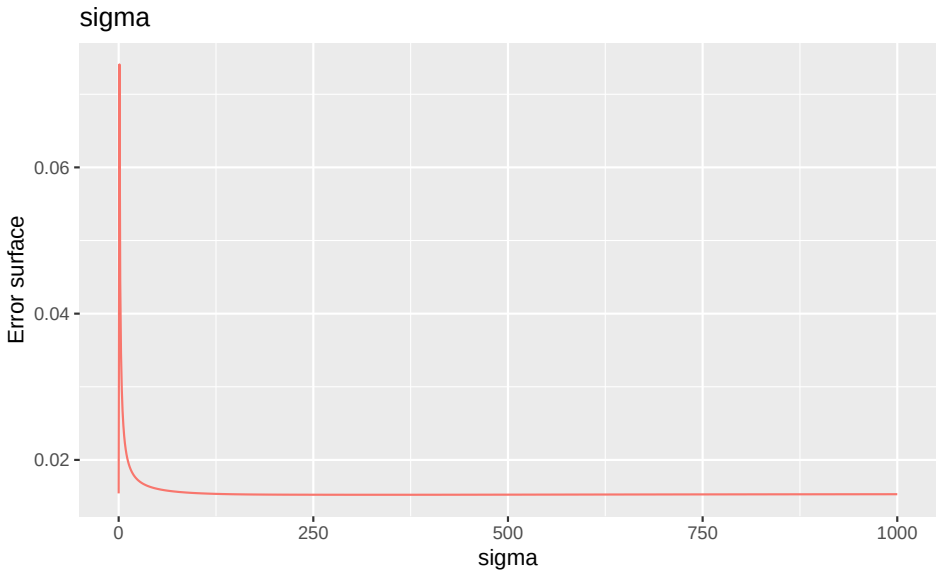
$$t_\theta(x) = \sum_{j=1}^{k} w_j \cdot [A_j + B_j t_{g,h}(x, g_j, h_j)]$$




Mixture proportion
Error surface
0.030
0.025
0.020
0.015
0.00
0.25
0.50
0.75
1.00
Mixture proportion


Lambda
Error surface
10
5
0
0
500
1000
1500
2000
Lambda


mu
Error surface
0.0155
0.0154
0.0153
0
500
1000
1500
2000
mu


sigma
Error surface
0.06
0.04
0.02
0
250
500
750
1000
sigma

System-under-test/test-system numbers

- Number of servers/appliances: 397
- Number of clusters: 119 (unique platform types 7)
- Number of functions: 368
- Number of business functions: 48
- Number of state-transitions in I/O FSM model: 915
- Number of third-parties being simulated: 7
- Number of tests included: 29

Test execution plan

Configuration:

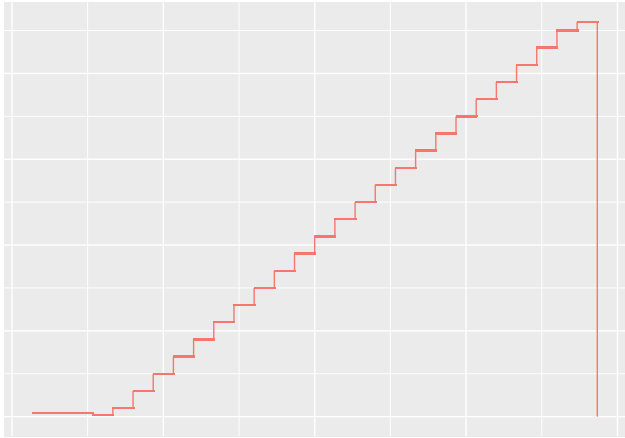
- Weights amongst business functions
- Idle-time distribution
- Think-time distributions
- Third-party response time distributions

Milestones:

- First milestone: required customer arrival rate
- Second milestone: nominal capacity provisioned for customer arrival rate

Load plan:

- Step load interval duration
- Step increment of simulated I/O FSMs.



Gathering Metric Data

Gathering metrics

Performance data:

- *Response times of functions hosted by system-under-test*
- *Outcomes of function calls — good, bad, fault, time-out, etc.*
- *Throughput rates — global and individual function call rates*

There are two resource classes of particular interest:

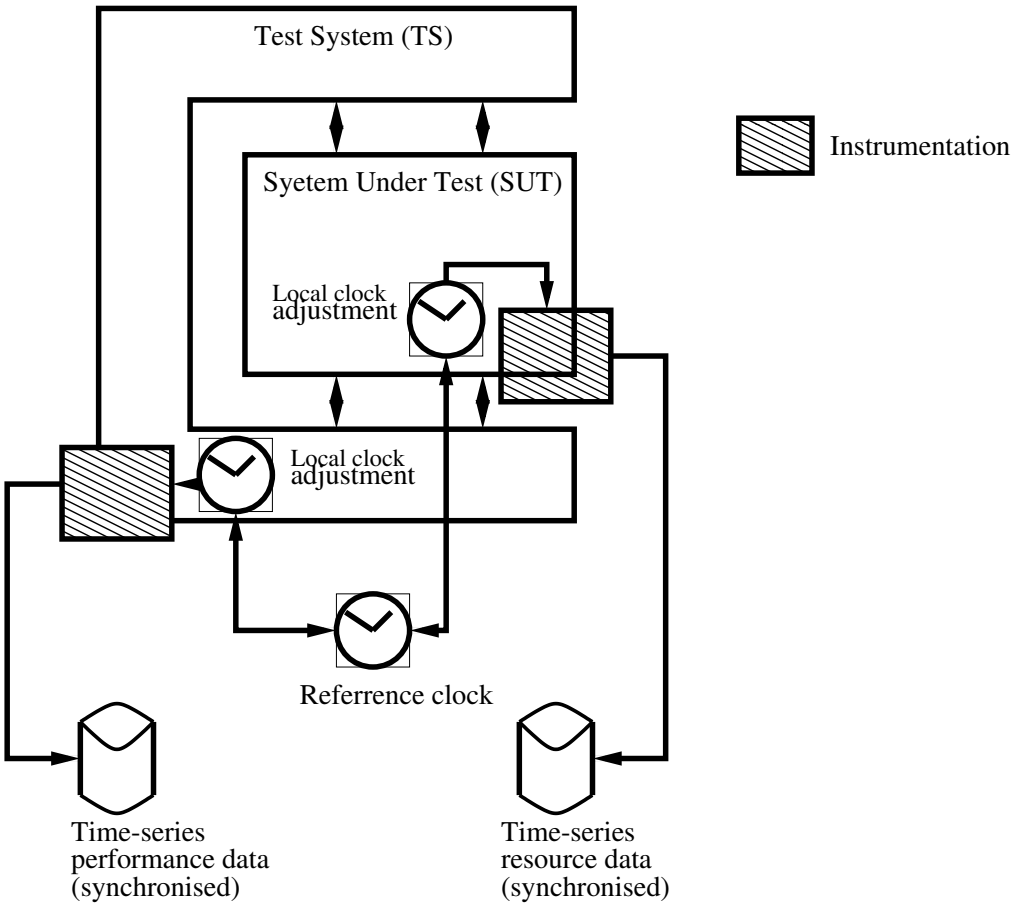
- *Percent of provisioned CPU capacity consumed*
- *Network bandwidth consumption*

Test-system and system-under-test metrics

$$\left[t_{k_i}, l, \sum_{j=1}^{k_i} n_j, \sum_{j=1}^{k_i} x_j, \sum_{j=1}^{k_i} x_j^2 \right]$$

$$\left[t_{k_1}, l, \sum_{j=k_1+1}^{k_2} n_j, \sum_{j=k_1+1}^{k_2} x_j, \sum_{j=k_1+1}^{k_2} x_j^2 \right]$$

- t_{k_i} is the timestamp of the k_i-th sample
- l is a vector of labels identifying the metric



Interval statistics

The mean and variance can be recovered from any recording interval, and any multiple of recording intervals including for each of the load intervals. For example, between timestamps t_{k_1} and t_{k_2}:

$$
\begin{aligned}
n_{t_{k_1},I} &= \sum_{j=k_1+1}^{k_2} n_j \\
m_{t_{k_1},I} &= \frac{1}{n_{t_{k_1},I}} \sum_{j=k_1+1}^{k_2} x_j \\
s^2_{t_{k_1},I} &= \left[\frac{1}{n_{t_{k_1},I}} \sum_{j=k_1+1}^{k_2} x_j^2 - \left(m_{t_{k_1},I}\right)^2 \right] \times \frac{n_{t_{k_1},I}}{n_{t_{k_1},I}-1} \\
s_{t_{k_1},I} &= \sqrt{s^2_{t_{k_1},I}}
\end{aligned}
$$

In the implementation, the summations are achieved by the component wise subtraction of values at time t_{k_1} from time t_{k_2}.

Checking Traits

Properties of the system-under-test: Testable

- The system-under-test is expected to consume resources in a manner linear in the load applied.
- The system-under-test is capable of a customer arrival rate that exceeds the first milestone rate.
- The system-under-test is capable of achieving a customer arrival rate equal to the second milestone rate.
- The response times of the individual functions are acceptable at the first milestone rate.
- The physical resource costs (in terms of percent provisioned CPU usage and network bandwidth usage) are acceptable.
- The background load does not overly consume the provisioned capacity (background load is the activity on a server or cluster that is not related to the offered load, that is, not related to the workload driven by the test-system).
- Clusters should be load balanced where provided for capacity and not just redundancy. Both CPU usage and network bandwidth usage are expected to be balanced.

Comparisons to previous tests

It is only sometimes possible to get performance and resource requirements upfront. Still, the system's behaviour in the field and under test often produces resource usage and performance precedents. Where past tests have been acceptable, the corresponding metrics can be used to test future tests for significant changes (degradation or improvement).

Possible comparisons:

- Change in performance - response times of a particular function call.
- Change in cost of a customer in terms of resources (CPU usage or network bandwidth usage).
- Change in background loads of the servers/clusters (CPU usage or network bandwidth usage).

Properties - testing and reporting

But comparisons need:

- Repeatable test-plans and configurations with identifying labels (minimising changes in conditions outside the system-under-test within the same configuration).
- Analysis arranged in a pipeline to ensure a repeatable process across repeated tests.
- Common physical resource usage units across heterogeneous landscape of the system-under-test.

Methods

Methods

Linear Regression (Simple Linear Regression - SLR): To model the relationship between system load (e.g., customer arrival rate) and resource usage (e.g., CPU, network).

Local Regression (LOESS): To model non-linear relationships in performance metrics (like response times) over time or load intervals.

Non-Parametric Bootstrap Testing: To compare current test metrics against historical tests without strong assumptions about underlying distributions.

Random Number Generation and Validation: Used to simulate user behaviour and variability in I/O FSM models.

Outlier Detection and Treatment: Improve model accuracy by removing poor-quality or high-influence observations.

Handling of Missing Data: Ensure analysis robustness in the presence of incomplete monitoring data.

Methods - Prepossessing

- Obfuscate customer identified sensitive data.
- Implementation of maximal length mixed linear congruential generator for pseudo random number generation.
- Implementation of uniform, normal (via NR Marsaglia polar method) and exponential pseudo random number functions in test-system (used in simulating idle-times, think-times, and third-party response times).
- Prepossessing the resource usage metrics to common units across heterogeneous components.
- Formulation of interval observations based on load intervals.
- Filter out poor quality observations and set aside for assignable cause determination.
- Filter out high influence observations and set aside for assignable cause determination.
- Determine if first and second milestones have been reached on remaining observations. Test fails if the first milestone is not reached.

Methods - Analysis

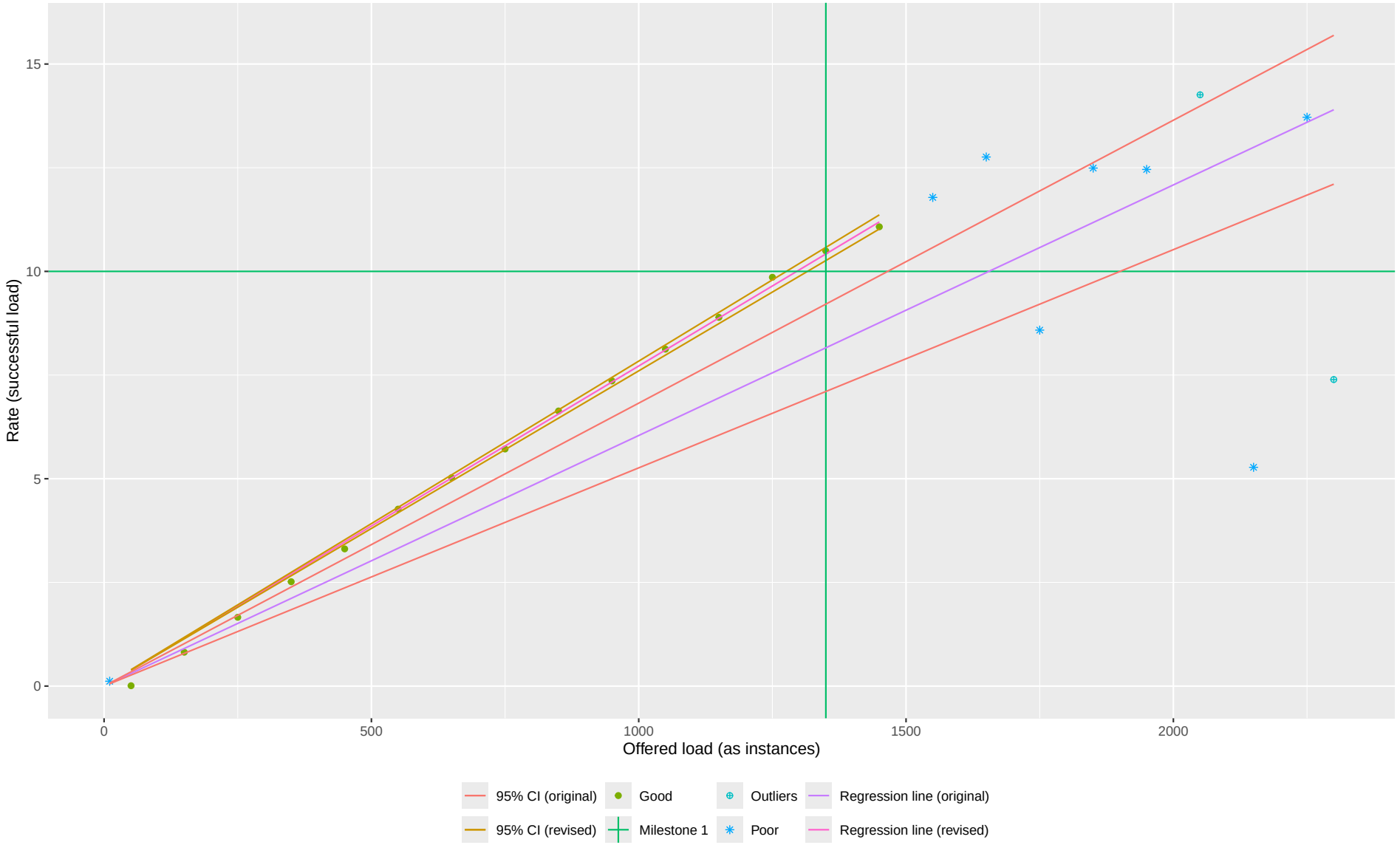
- SLR: Regression through the origin to find deviations from offered and successful load.
- SLR: Regress resource usage onto customer arrival rate across remaining observations: This produces load independent and load dependent levels of resource usage.
- Local regression: To capture non-linear relationship between response time and customer arrival rate, and to estimate the response time at the first milestone rate, and including a confidence interval.
- Non-parametric bootstrap test: Difficult to pool the linear model parameter estimates over previous tests; and pooling of response times at first milestone not appropriate for a t-test since the population is not normal (typically significantly right skewed), and cannot claim to have a common variance.

Methods - Presentation

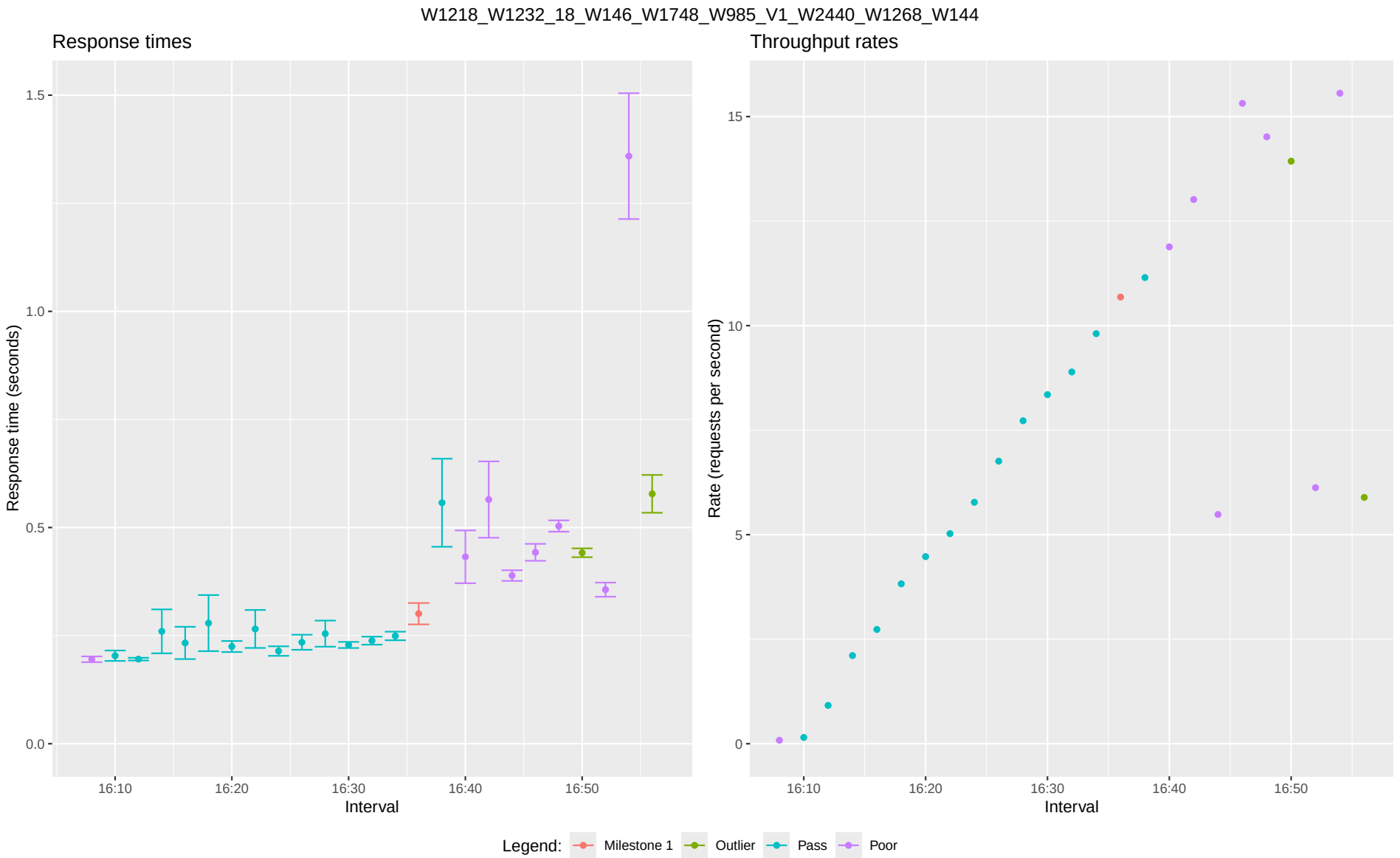
- Small multiples (after Tufte) as proof sheets:
  - Common scales and arranged in a grid makes it easy to compare estimates across the clusters.
  - For resources, cluster members on the same plot makes it easy to see imbalances in estimates for both load independent usage and load dependent usage.
- Comparison to previous tests:
  - Detect and report on changes in performance.
  - Detect and report on changes in resource usage dependent on load.
  - Detect and report on changes in resource usage independent of load.

Example

Poor quality and high-influence observations



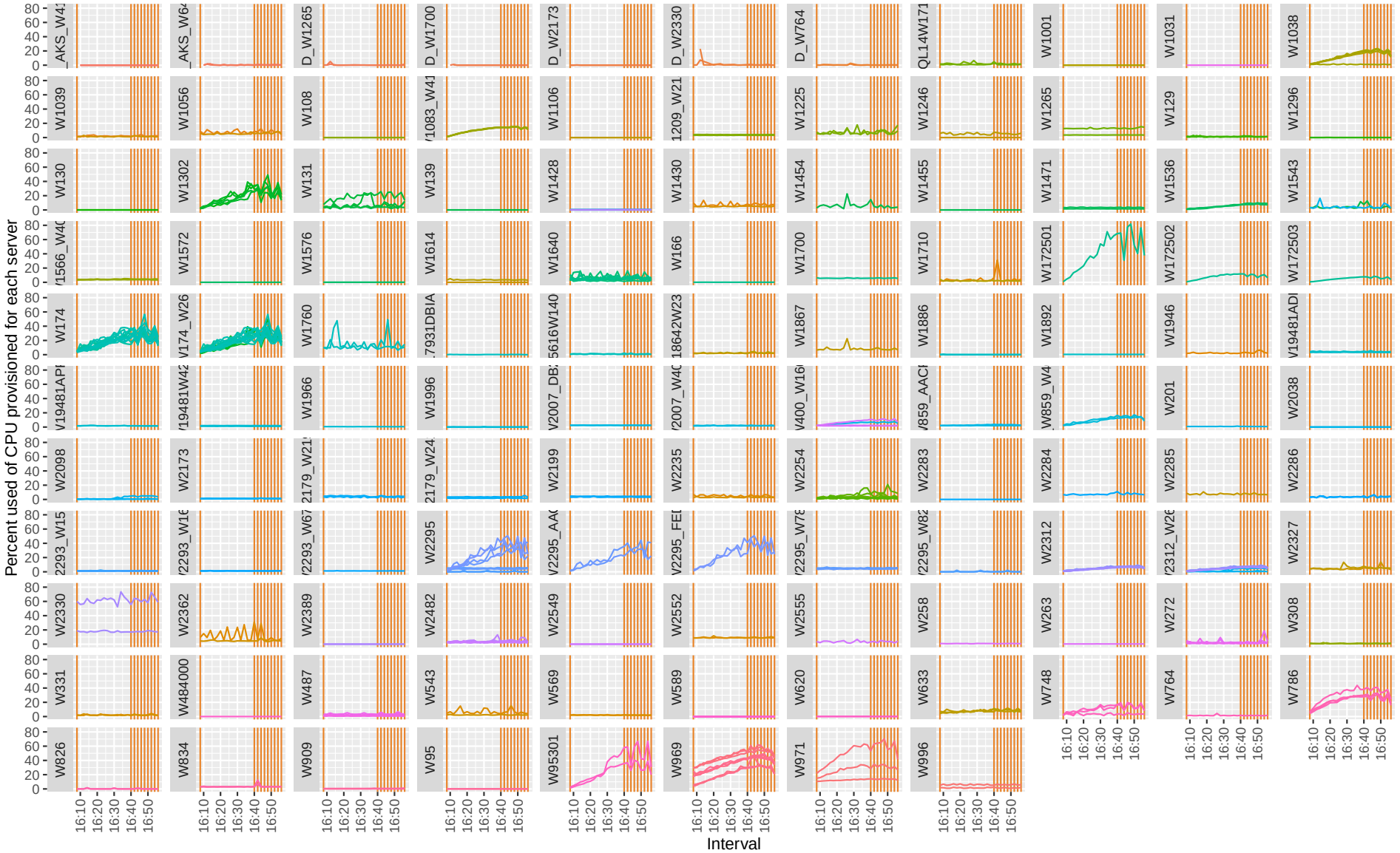
Performance - Response times and throughput



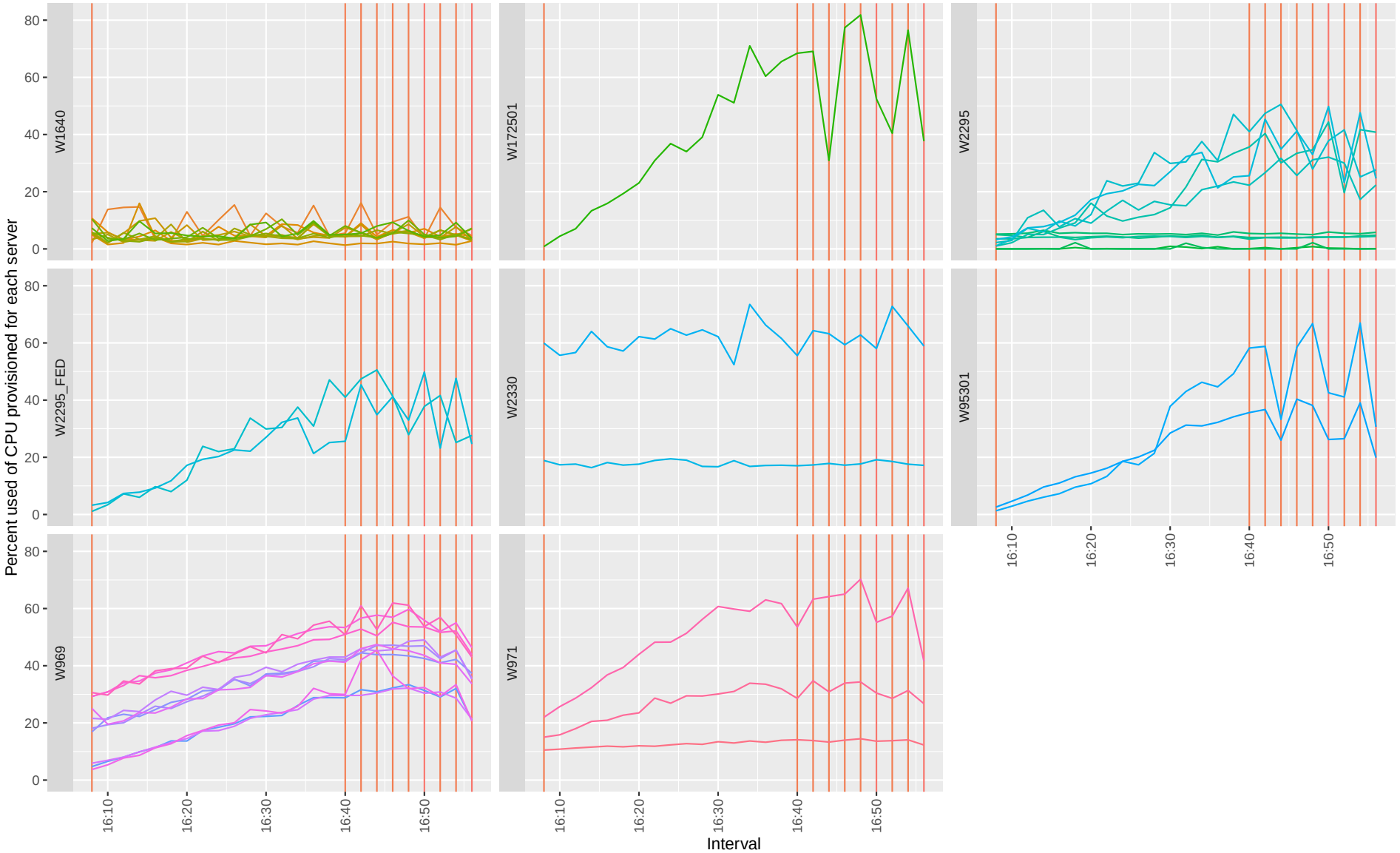
Fitted response times at first milestone

Test Number	Basename	Local Regression Predict	Local Regression StdErr	Local Regression 95CILo	Local Regression 95CIHi	Samp Size	Samp Mean Resp	Samp Var Resp	SEM	Rate
1	W1218	11.814	0.713	10.327	13.302	1301	10.202	0.308	0.015	10.842
1	W1237	11.696	0.733	10.146	13.245	8	9.875	0.113	0.119	0.067
1	W999_	11.585	0.522	10.453	12.718	4	10.258	0.099	0.158	
1	W999_	11.486	0.642	10.105	12.866	3	10.333	0.374	0.353	0.025
1	W1822	11.447	0.599	10.192	12.701	84	10.182	0.354	0.065	0.700
1	W1822	11.332	0.554	10.173	12.491	282	10.147	0.292	0.032	2.350
1	W1822	11.302	0.569	10.107	12.498	281	10.109	0.288	0.032	2.342
1	W1237	11.273	0.541	10.136	12.411	71	10.063	0.256	0.060	0.592
1	W1822	11.205	0.505	10.149	12.262	76	9.995	0.269	0.059	0.633
1	W999_	10.367	0.448	9.358	11.376	4	9.736	0.094	0.154	0.033
1	W1237	3.770	0.173	3.406	4.134	65	3.415	5.809	0.299	0.542
1	W999_	3.701	0.247	3.164	4.238	3	2.970	0.002	0.029	0.025
1	W999_	3.679	0.258	3.117	4.240	1	3.349			0.008
1	W999_	3.667	0.378	2.797	4.537	5	3.070	0.011	0.046	0.042
1	W1237	2.738	0.178	2.357	3.119	6	2.391	0.039	0.081	0.050
1	W1237	2.058	0.124	1.793	2.324	5	2.345	1.780	0.597	0.042
1	W1237	1.932	0.088	1.747	2.118	70	1.726	0.196	0.053	0.583
1	W1218	1.849	0.088	1.665	2.033	1277	1.632	0.544	0.021	10.642
1	*W121	0.536	0.085	0.359	0.713	1282	0.301	0.203	0.013	10.683

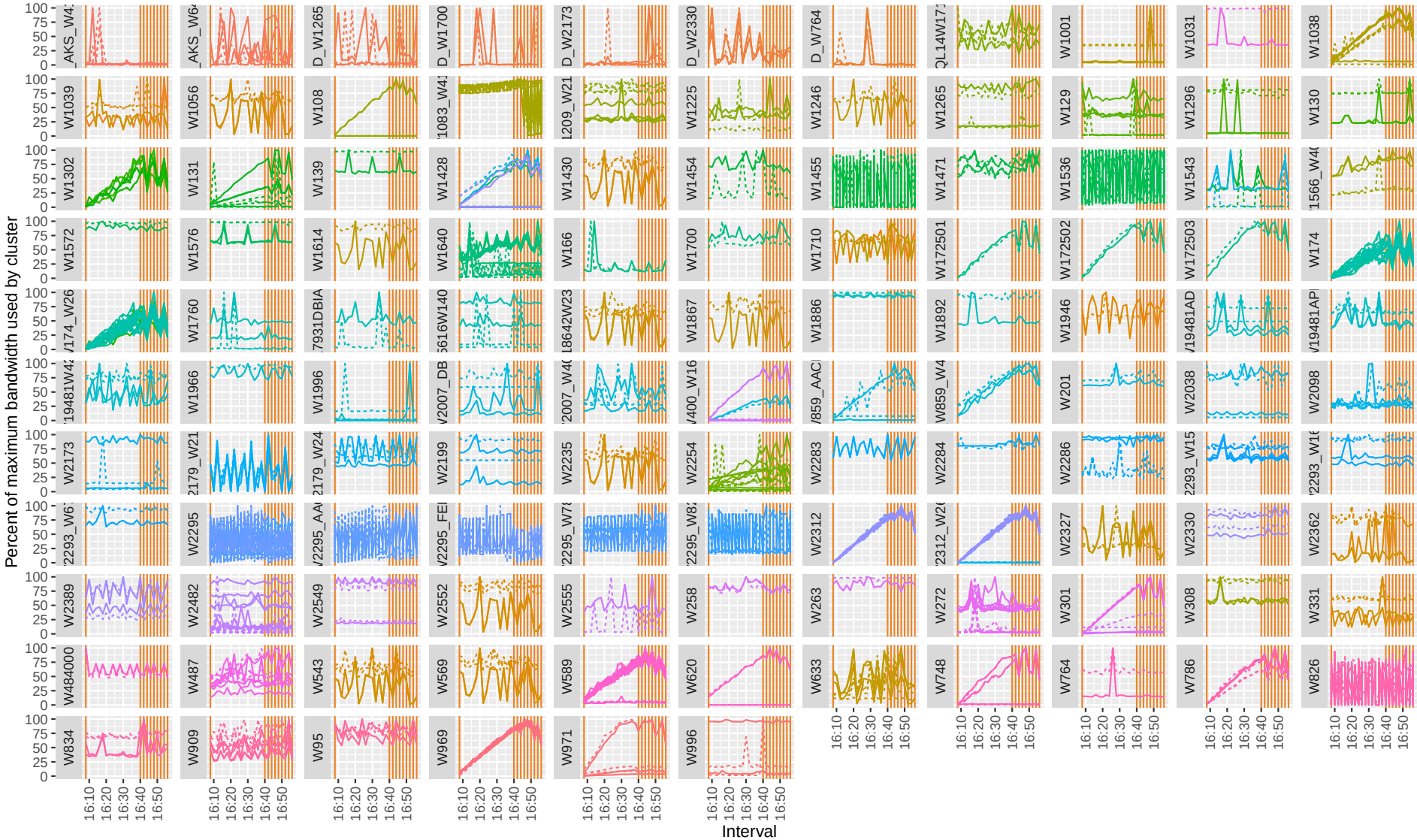
Server/cluster percent CPU usage



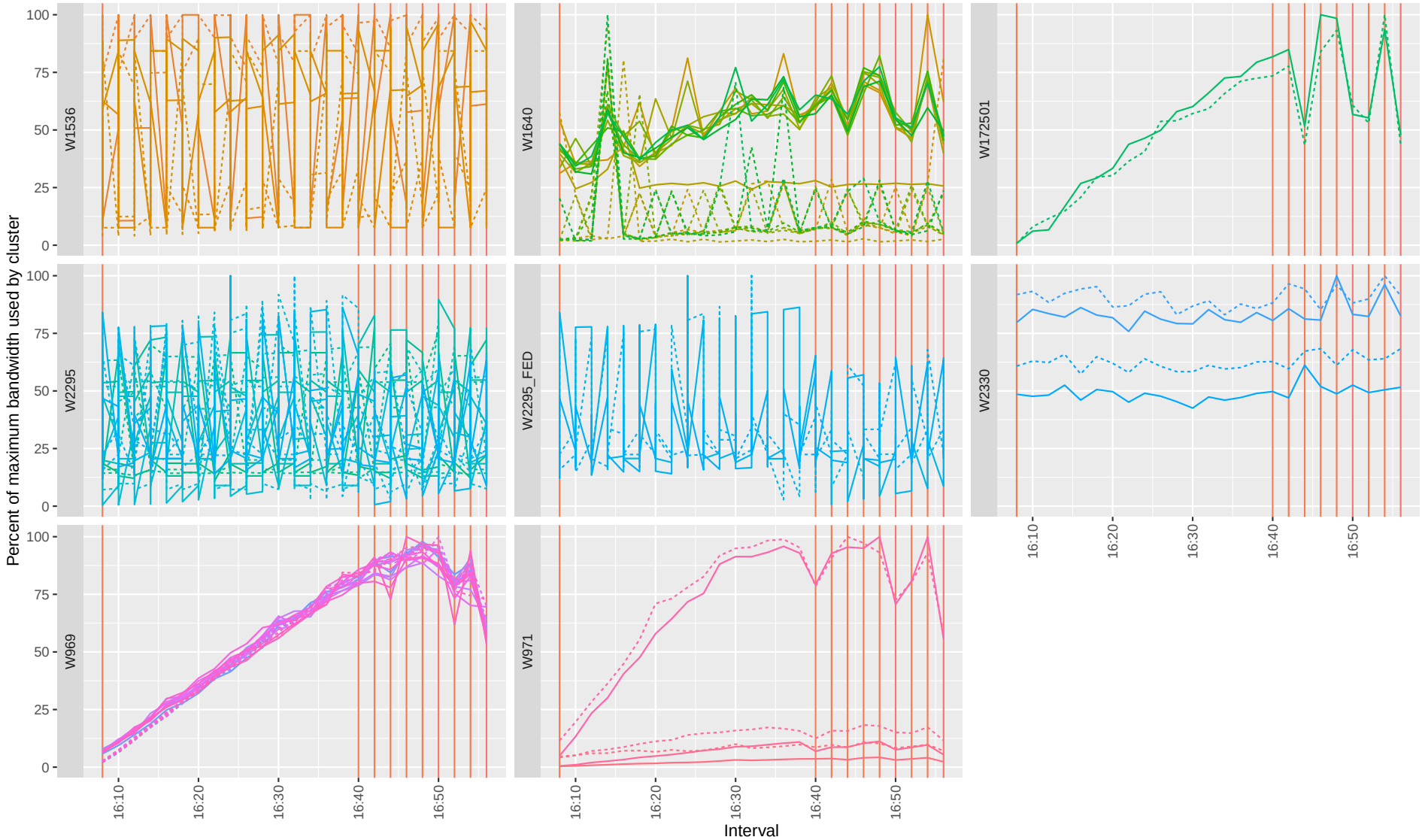
Server/cluster percent CPU usage - Observations



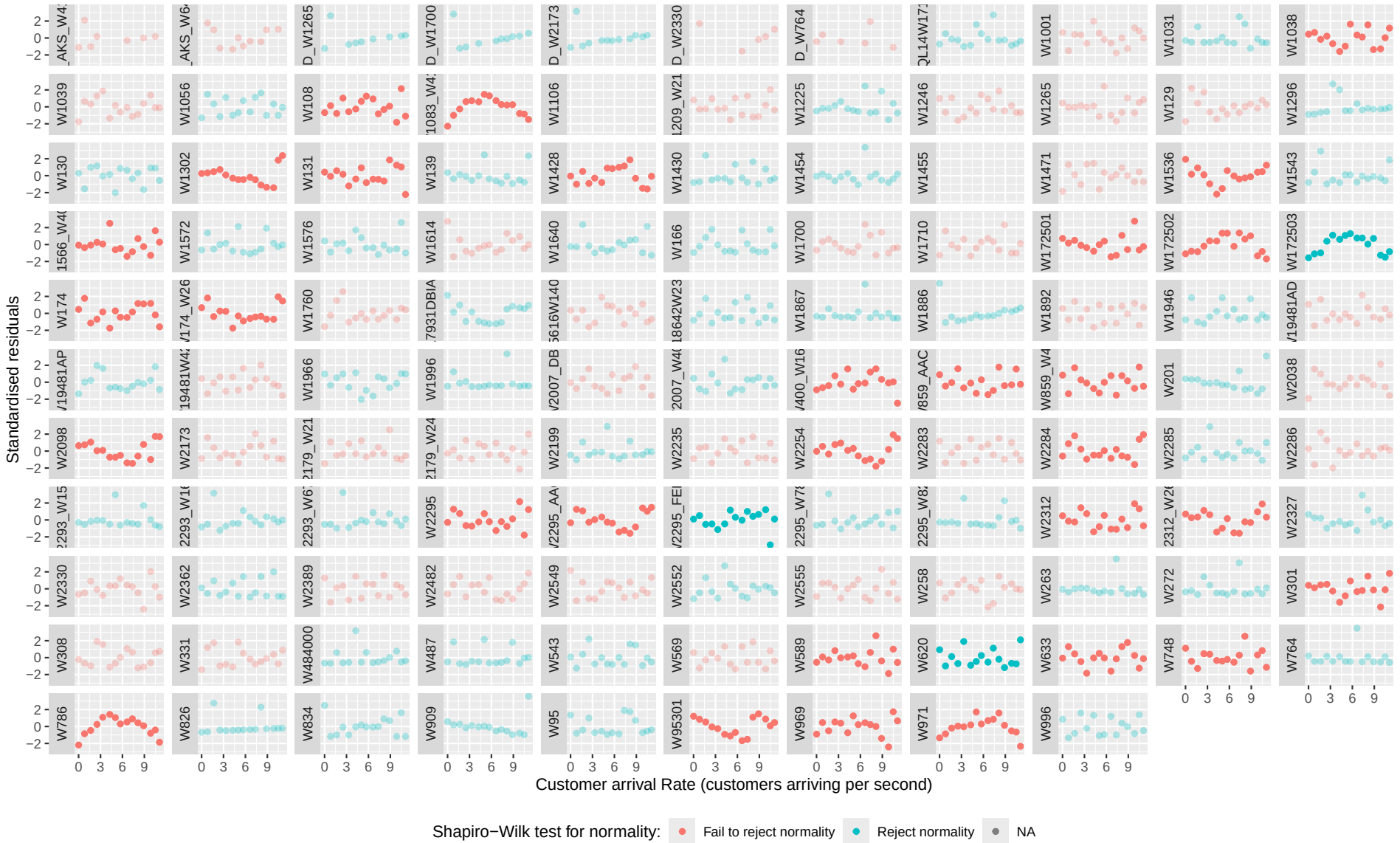
Server/cluster network bandwidth usage

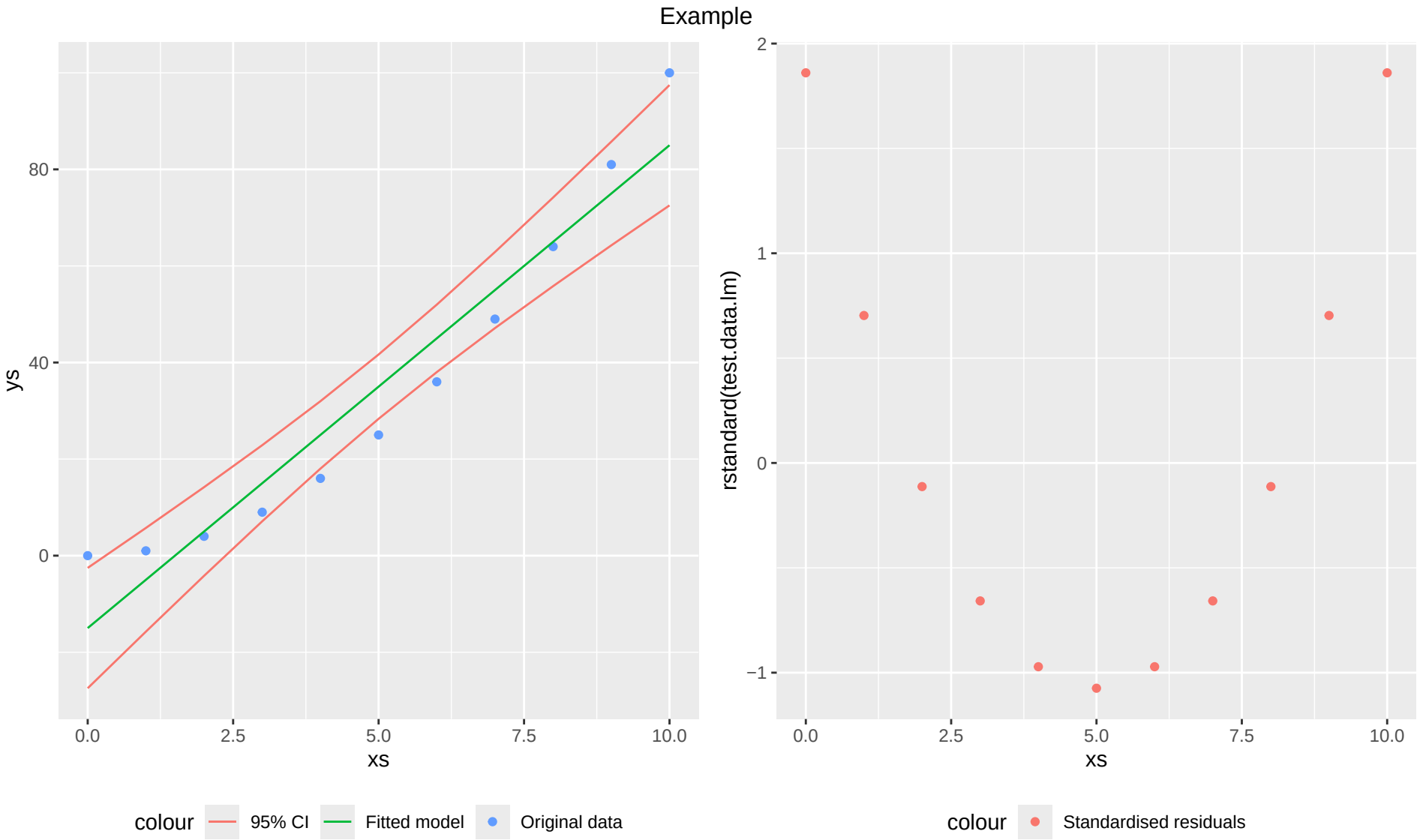


Server/cluster network bandwidth usage - Observations

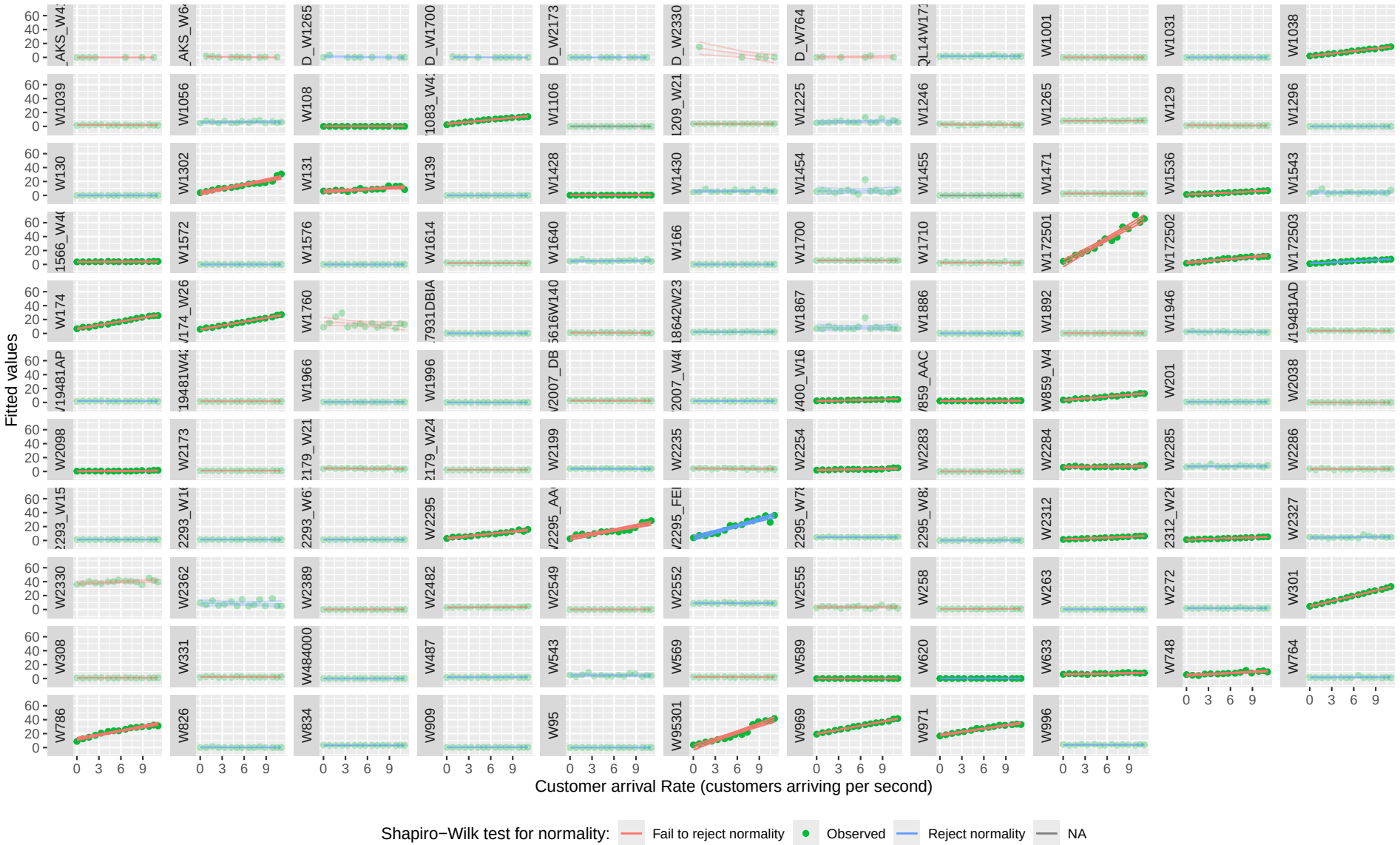


Diagnostics - Standardised residuals for CPU usage




Example
80
40
0
ys
0.0
2.5
5.0
7.5
10.0
xs
colour
95% CI
Fitted model
Original data
2
1
0
−1
rstandard(test.data.lm)
0.0
2.5
5.0
7.5
10.0
xs
colour
Standardised residuals

Fitted linear models for CPU usage

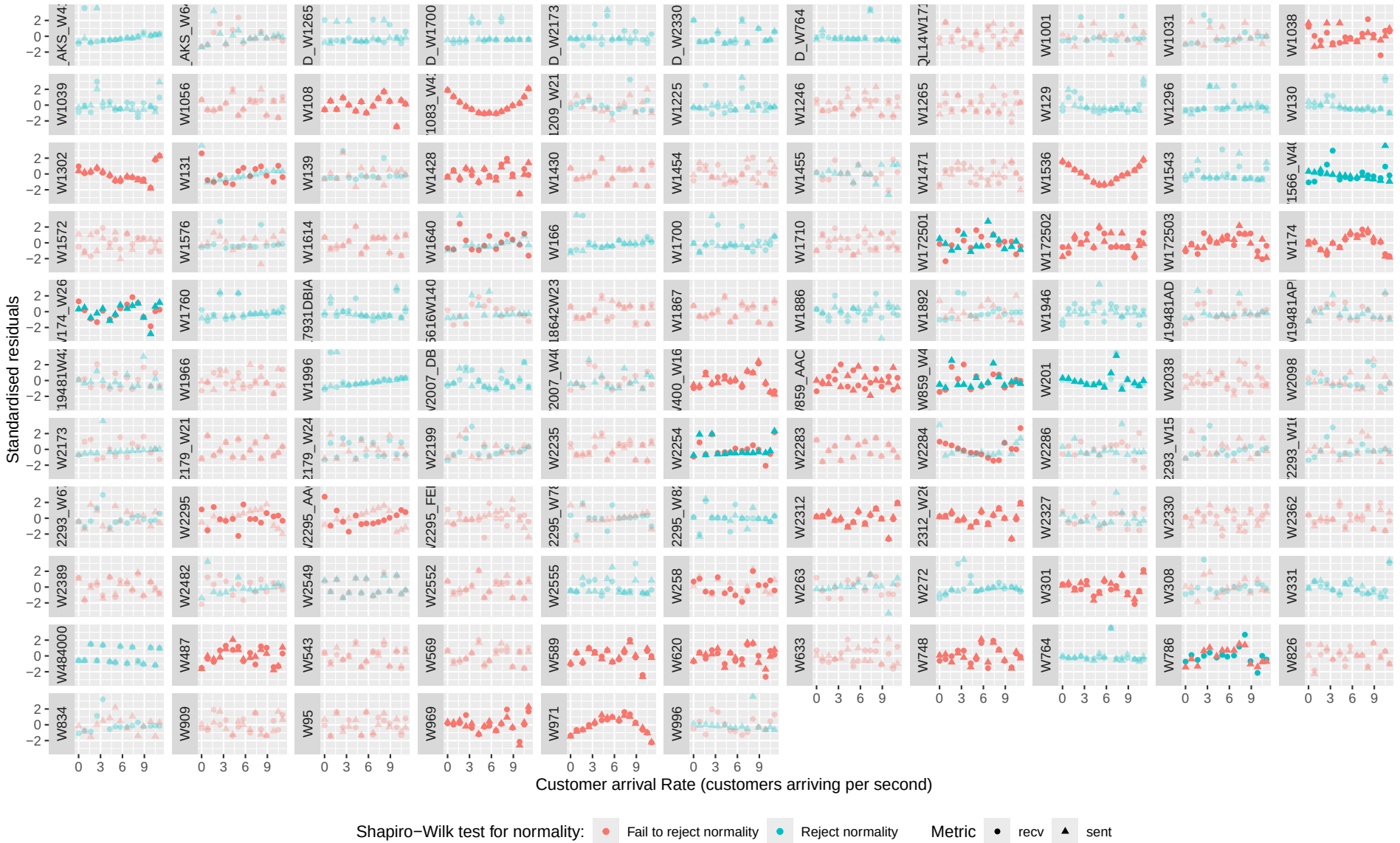


Estimated CPU usage considered excessive

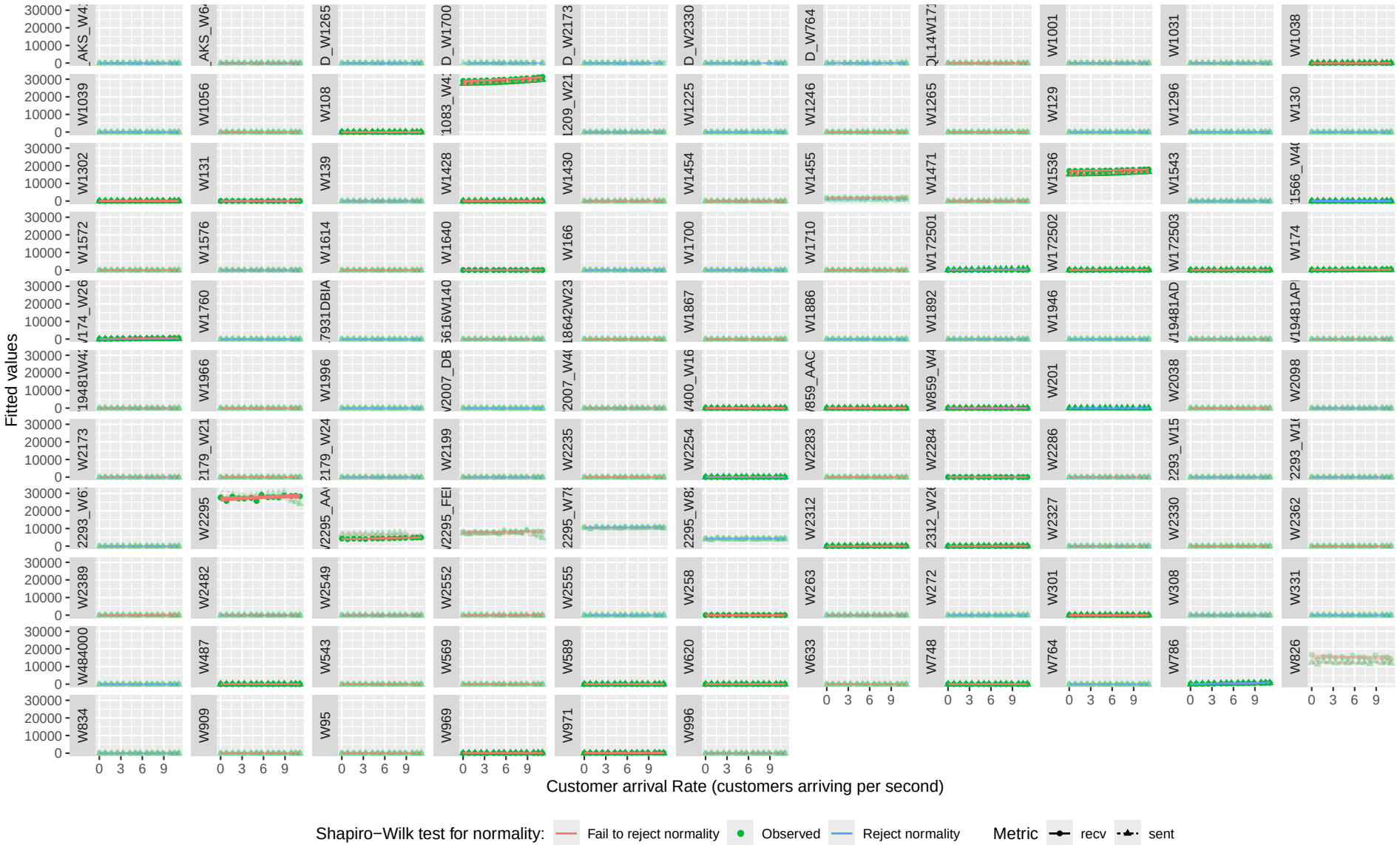
Cluster	Int Est	Int SE	Int t-val	Int p-val	Slope Est	Slope SE	Slope t-val	Slope p-val	R2	F-stat	df2	p-val	MS1 Est	MS2 Est
D_W2330	14.40	3.08	4.67	0.02	-1.49	0.37	-4.00	0.03	0.84	15.99	3	0.03	-0.46	-15.32
W172501	1.51	2.30	0.66	0.52	5.87	0.34	17.06	0.00	0.96	290.89	13	0.00	60.18	118.86
W1760	16.88	2.87	5.88	0.00	-0.56	0.43	-1.30	0.22	0.12	1.69	13	0.22	11.29	5.70
W2330	37.81	1.12	33.81	0.00	0.32	0.17	1.92	0.08	0.22	3.69	13	0.08	41.02	44.23
W786	12.25	0.91	13.51	0.00	1.97	0.14	14.52	0.00	0.94	210.89	13	0.00	31.96	51.67
W969	19.57	0.24	83.14	0.00	1.96	0.04	55.72	0.00	1.00	3104.95	13	0.00	39.20	58.83
W971	17.79	0.51	35.20	0.00	1.57	0.08	20.77	0.00	0.97	431.41	13	0.00	33.50	49.21

Estimated CPU capacity used at second milestone exceeds provisioned capacity and estimated background CPU capacity considered excessive.

Diagnostics - Std residuals for network usage



Fitted network bandwidth usage



Non-parametric bootstrap test

1. There are few assumptions for the bootstrap, and these can reasonable be assumed to hold in the present case.
2. The given sample is representative of the population.
3. Larger samples are preferred, and lead to more accurate bootstrap results. However, the bootstrap still works with small sample sizes.
4. A confidence interval can be determined from the quantiles of the bootstrap estimates or from a z-score and the sample variance of the bootstrap samples if $\widehat{\theta}$ is assumed to be approximately normal. To test the null-hypothesis that a give point is drawn from the same underlying population as the sample $\mathbf{x}$, the test proceeds by checking whether the point is outside of the confidence interval, in which case evidence at the prescribed α level of significance results in the rejection of the null-hypothesis. An estimated p-value can also be determined by assessing the proportion of the $\widehat{\theta}_i^*$ at least as extreme as $\widehat{\theta}$.

If T is the statistic of interest, then the bootstrap proceeds as follows for the given sample $\mathbf{x} = \{x_1, x_2 \ldots, x_n\}$, and where B is the number of times the given sample is resampled:

1. Compute the test statistic for the given sample: $\widehat{\theta} = T(\mathbf{x})$.
2. For $i = 1, \ldots, B$, resample $\mathbf{x}$ (sample with replacement), $\mathbf{x}^* = \{x_1^*, x_2^*, \ldots, x_n^*\}$ (called the bootstrap sample or resample), and compute $\widehat{\theta}_i^* = T(\mathbf{x}^*)$. The values $\widehat{\theta}_i^*$ are the bootstrap estimates.
3. The bootstrap estimate for θ is then $\overline{\widehat{\theta^*}} = \frac{1}{B} \sum_{i=1}^{B} \widehat{\theta}_i^*$.

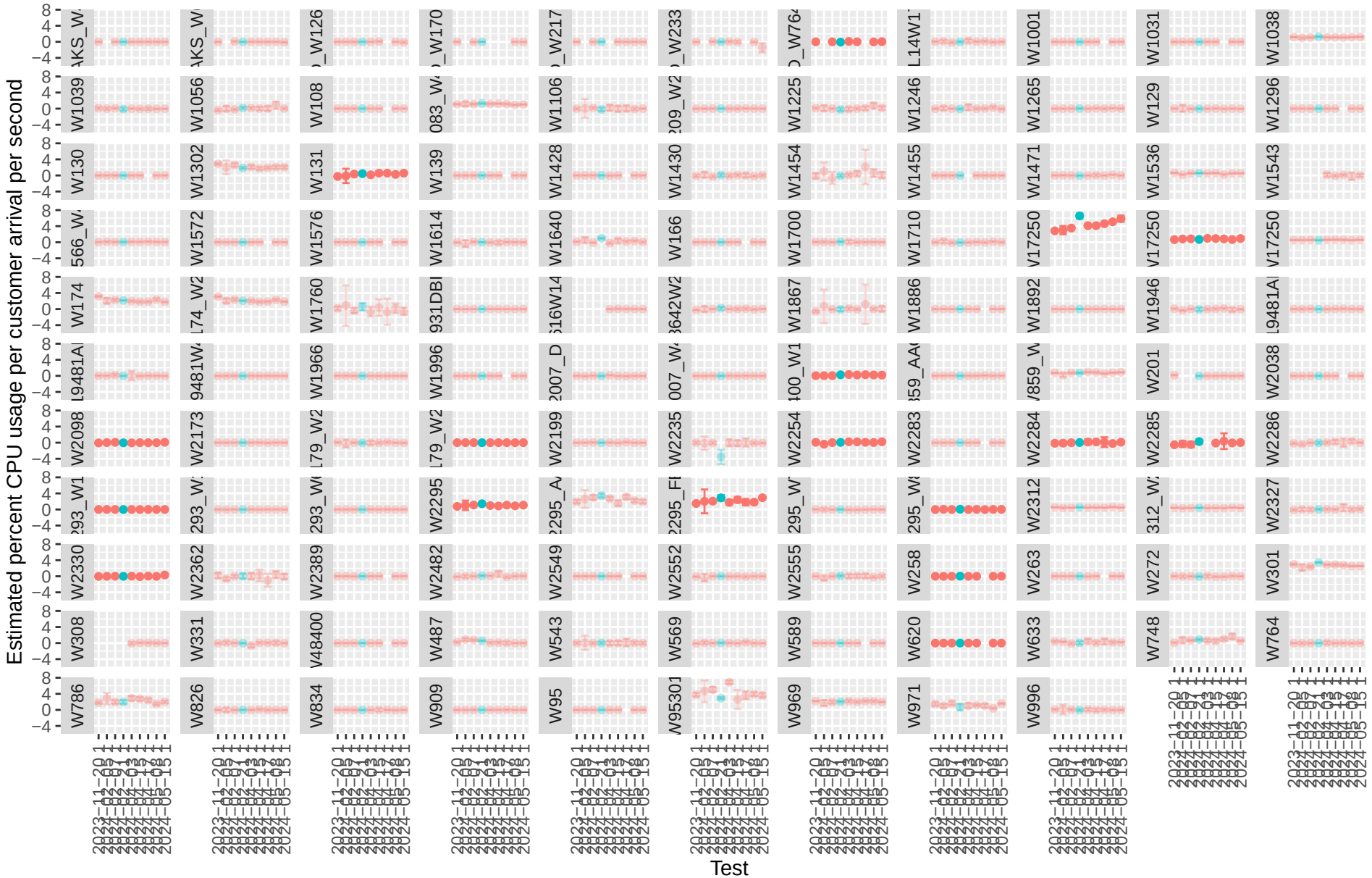
For the purposes of this study, the test statistic is the absolute value of the distance between the point being tested, y, and the mean of a sample $\mathbf{x}$, with the number of bootstrap samples in each test set to $B = 1000$

$$T(\mathbf{x}, y) \quad = \quad |\overline{\mathbf{x}} - y|$$

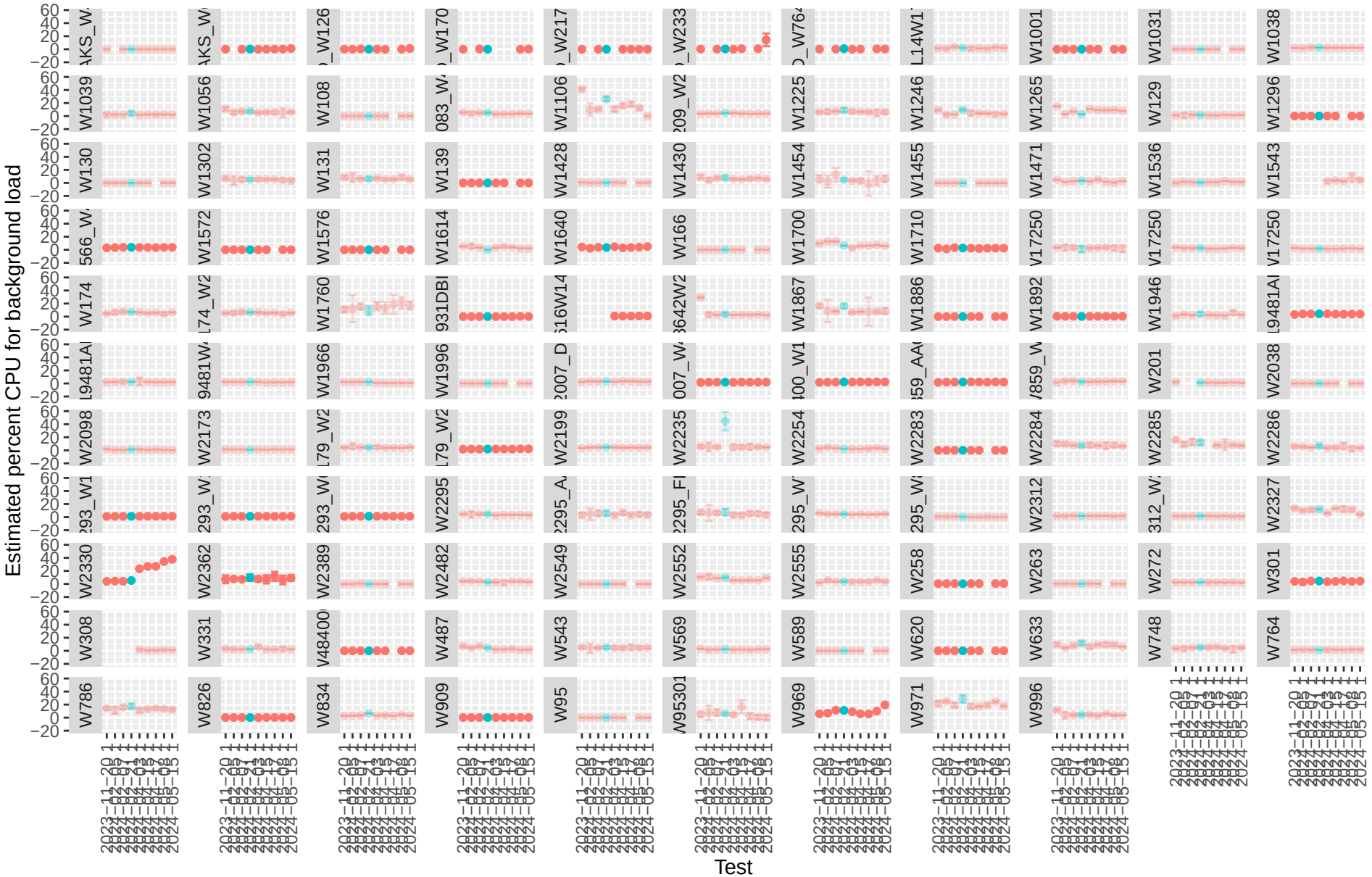
Bootstrap tests: Performance comparisons across tests

Test Number	Basename	Local Regression Predict	Local Regression 95CILo	Local Regression 95CIHi	Samp Size	Samp Mean Resp	Samp Var Resp	SEM	Rate	Bootstrap p-value
1	W1218	11.814	10.327	13.302	1301	10.202	0.308	0.015	10.842	0.000
1	W1237	11.696	10.146	13.245	8	9.875	0.113	0.119	0.067	0.003
1	W999_	11.585	10.453	12.718	4	10.258	0.099	0.158		0.000
1	W999_	11.486	10.105	12.866	3	10.333	0.374	0.353	0.025	0.000
1	W1822	11.447	10.192	12.701	84	10.182	0.354	0.065	0.700	0.000
1	W1822	11.332	10.173	12.491	282	10.147	0.292	0.032	2.350	0.003
1	W1822	11.302	10.107	12.498	281	10.109	0.288	0.032	2.342	0.000
1	W1237	11.273	10.136	12.411	71	10.063	0.256	0.060	0.592	0.008
1	W999_	1.504	0.729	2.280	5	1.748	2.128	0.652	0.042	0.004
1	W999_	1.502	1.209	1.795	3	1.613	0.278	0.304	0.025	0.000
1	W1237	1.409	1.042	1.777	5	1.412	0.214	0.207	0.042	0.028
1	W1822	1.375	1.076	1.674	558	1.052	0.142	0.016	4.650	0.000
1	W1822	1.365	1.044	1.686	548	1.039	0.156	0.017	4.567	0.001
1	W1822	1.355	1.055	1.655	169	1.070	0.169	0.032	1.408	0.000
1	W1822	1.352	1.086	1.617	153	1.070	0.196	0.036	1.275	0.003
1	W1237	1.329	1.124	1.533	65	1.126	0.194	0.055	0.542	0.027
1	W999_	1.178	0.778	1.578	1	0.780			0.008	0.000
1	W999_	1.153	0.573	1.733	4	0.399	0.067	0.129		0.000

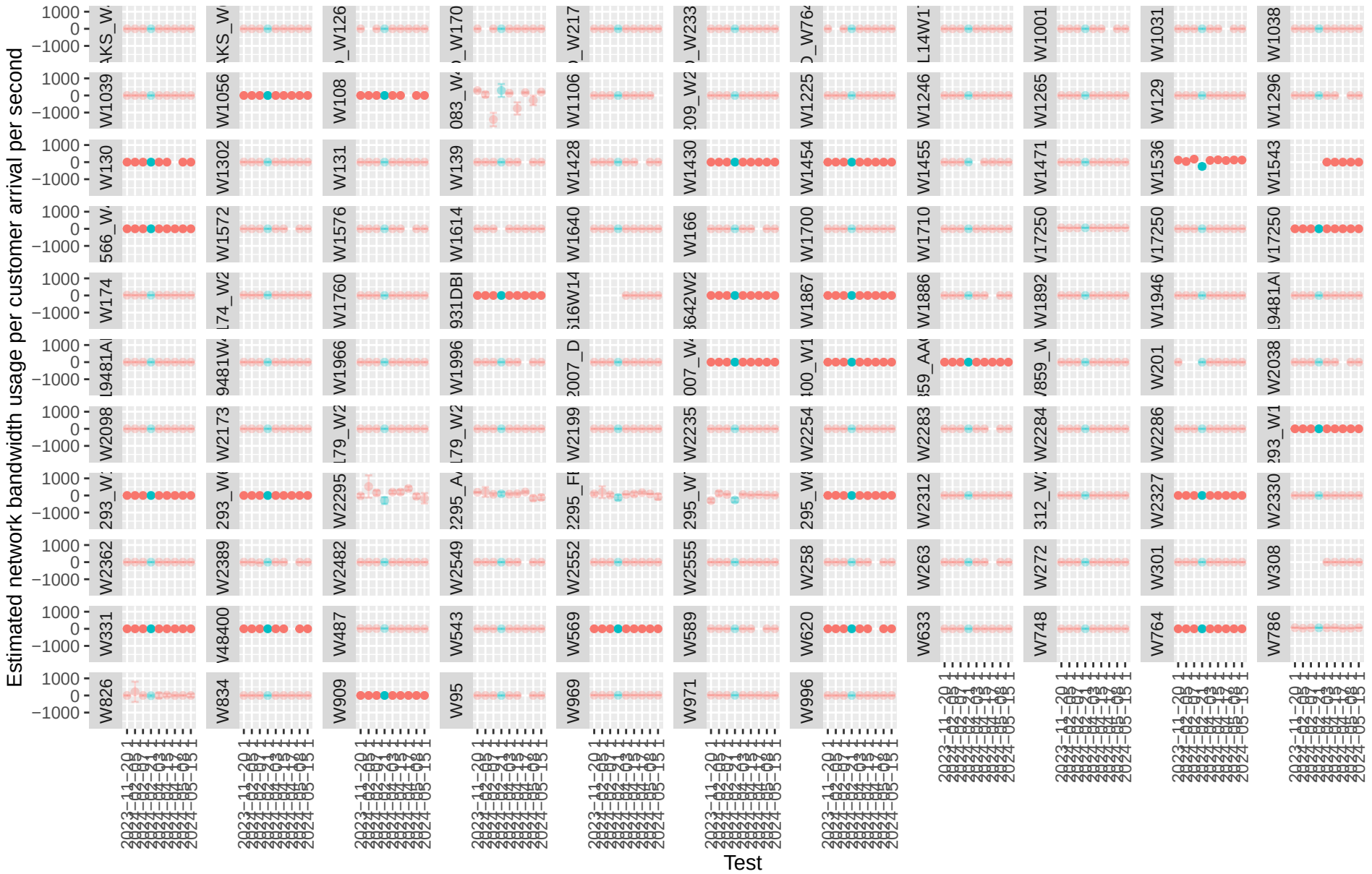
Load dependent CPU resource usage - slope



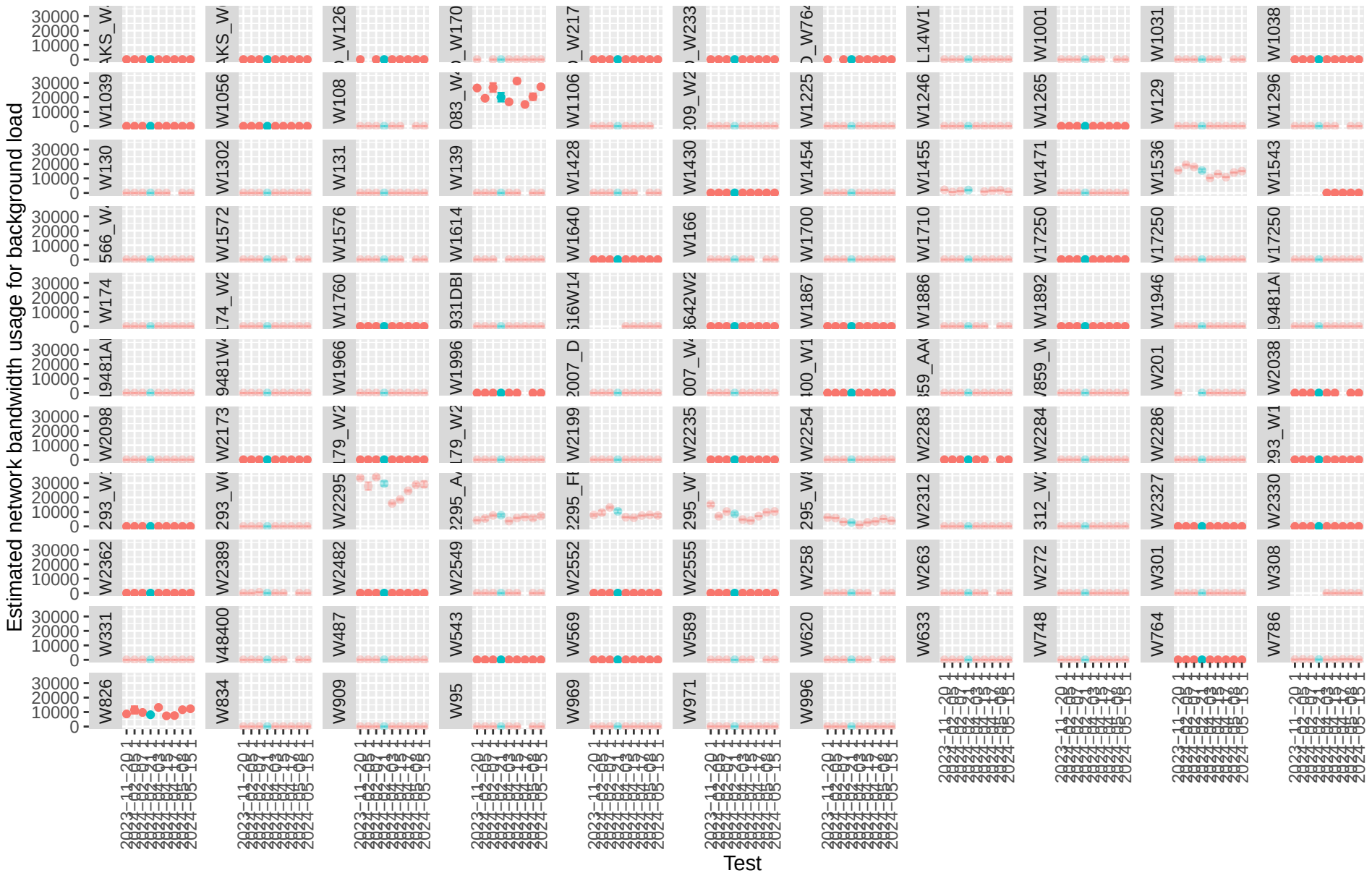
Load independent CPU resource usage - intercept



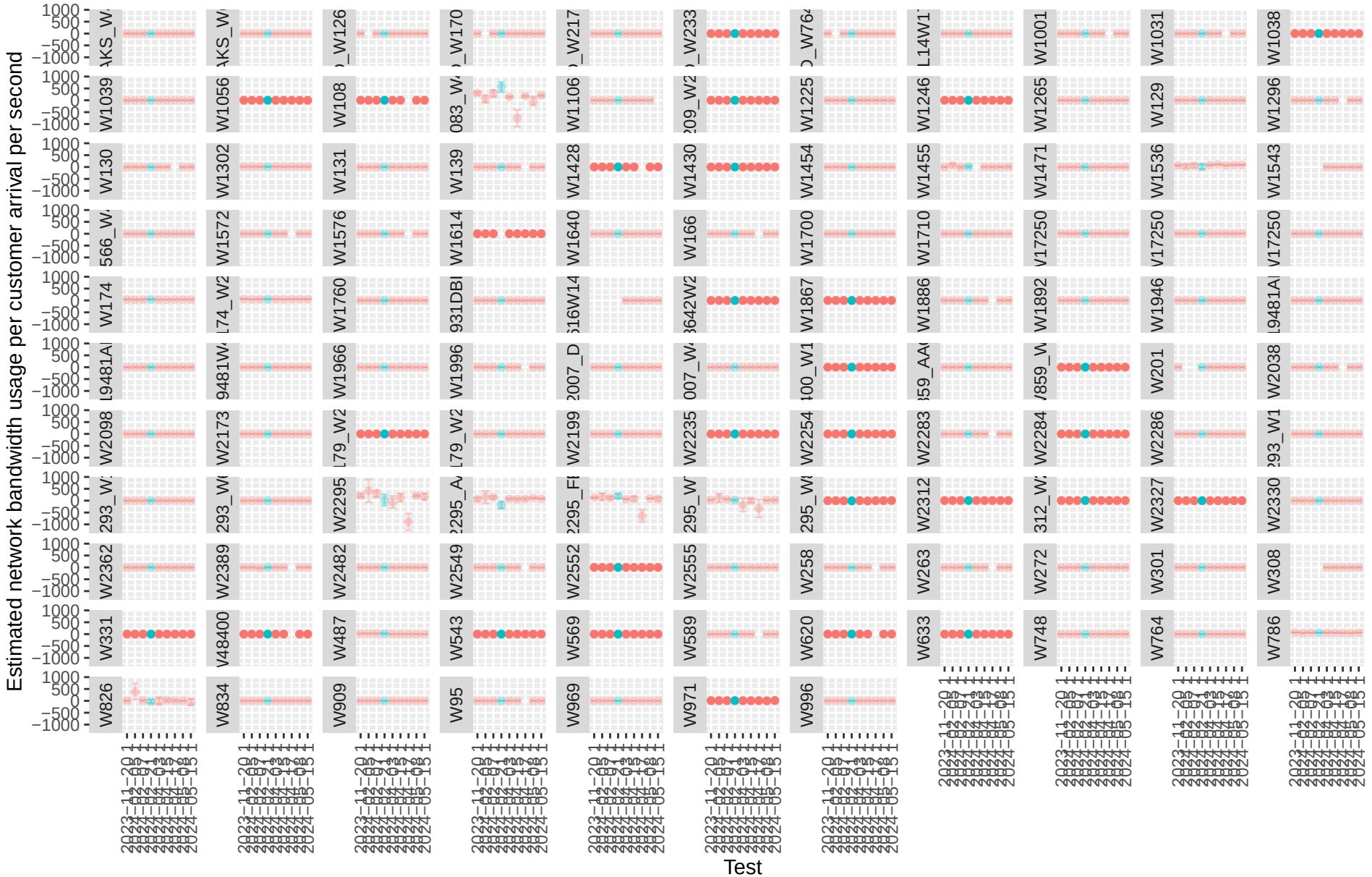
Load dependent network usage sent - slope



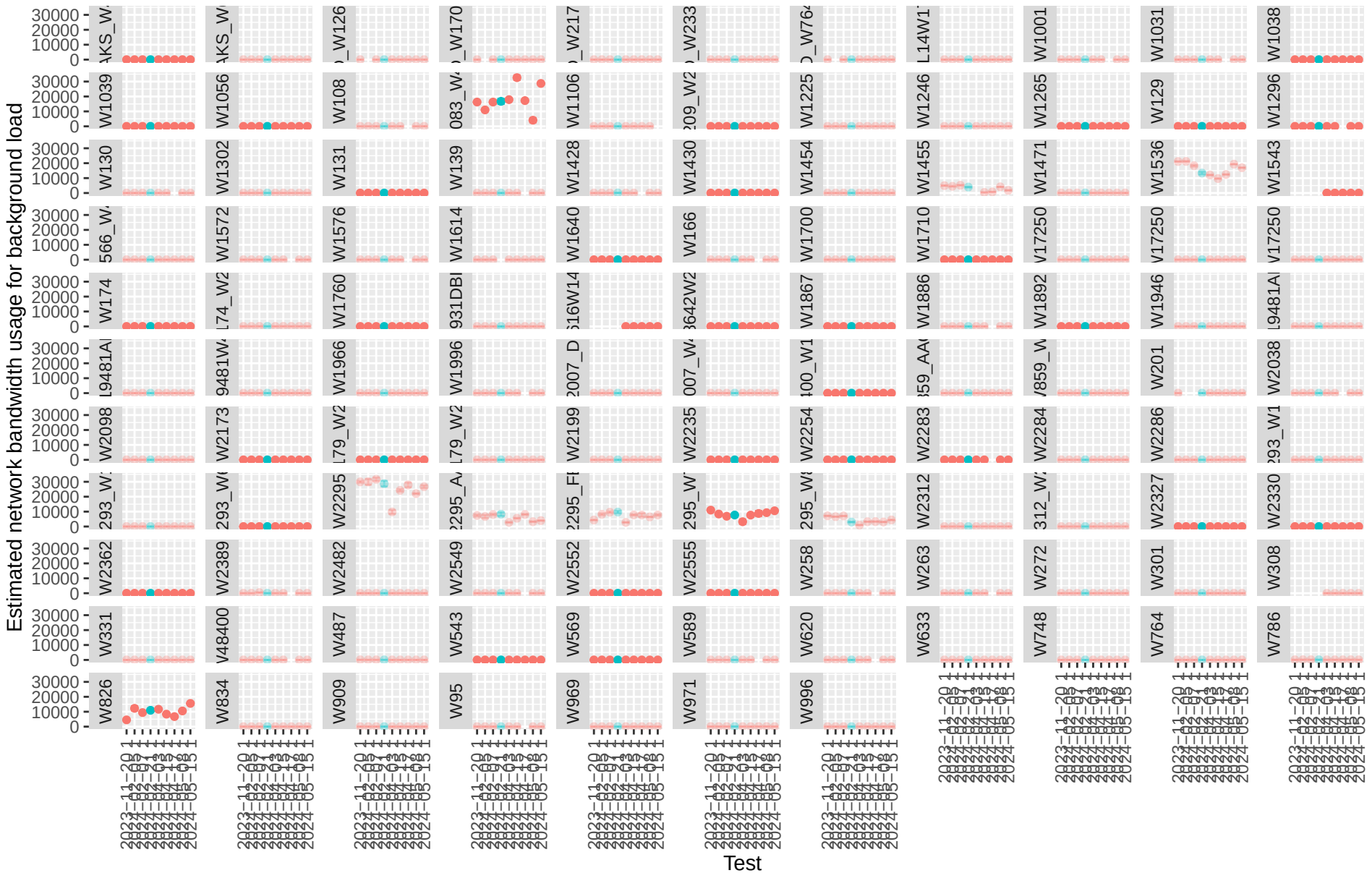
Load independent network usage sent - intercept



Load dependent network usage received - slope



Load independent network usage received - intercept

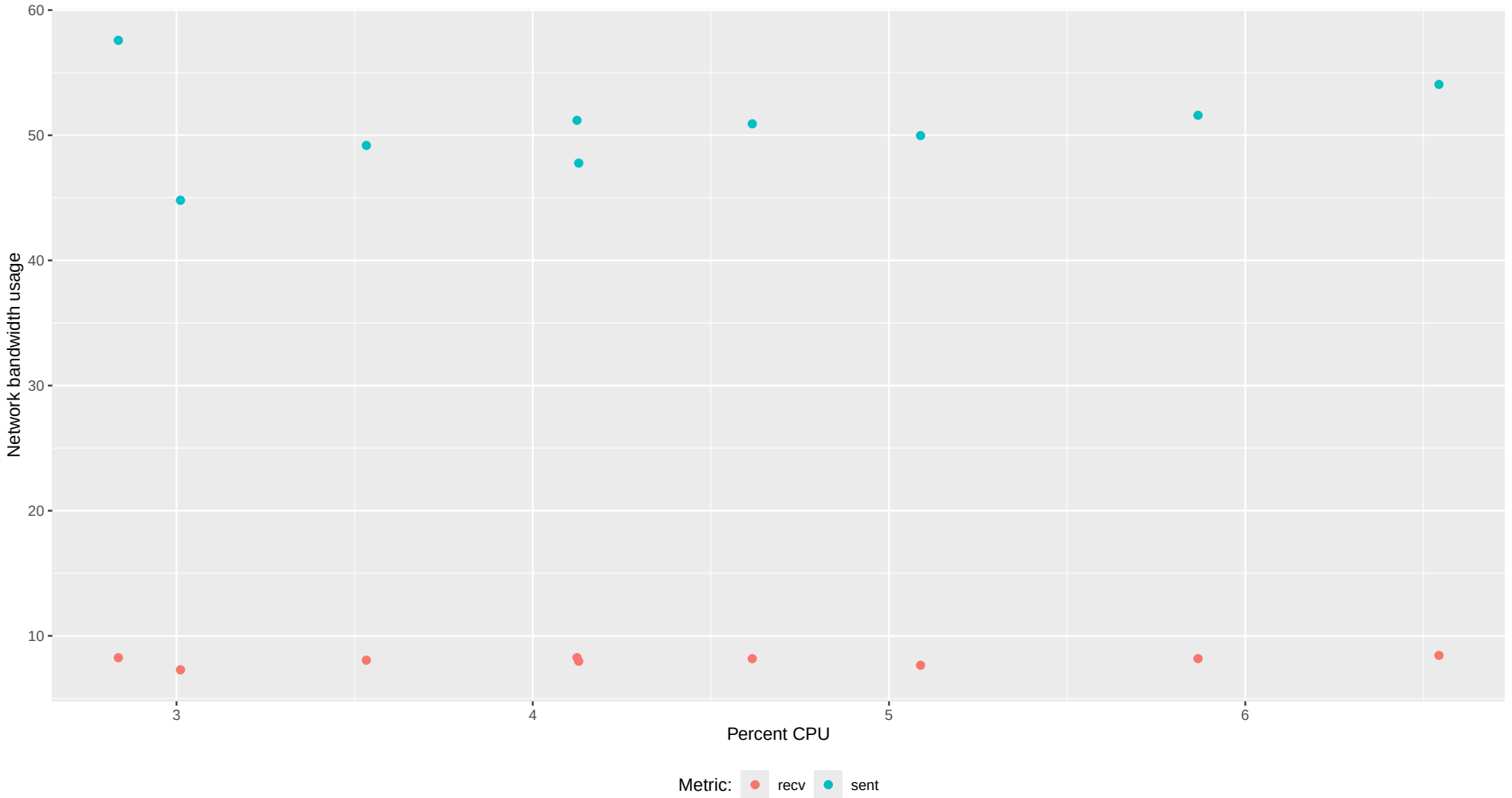


W172501 behaviour across tests

Regressing CPU usage estimate onto the network estimate metrics for cluster W172501:

Metric	Slope Est	Slope Std Err	Slope t-val	Slope p-val	R2	F Stat	df2	p-val
sent	0.07	0.13	0.55	0.60	0.04	0.31	7	0.60
recv	1.46	1.21	1.21	0.27	0.17	1.46	7	0.27

W172501 behaviour across tests - First milestone estimates



Testing for imbalance

There is some noise in the load balancing, nevertheless it is possible to test balancing with a statistical test on the slope coefficients of the resource usage fitted models. Under the balance hypothesis:

$$t^* = \frac{\widehat{\beta}_1 - 1}{\mathrm{SE}(\widehat{\beta}_1)} \quad \text{for testing CPU usage}$$

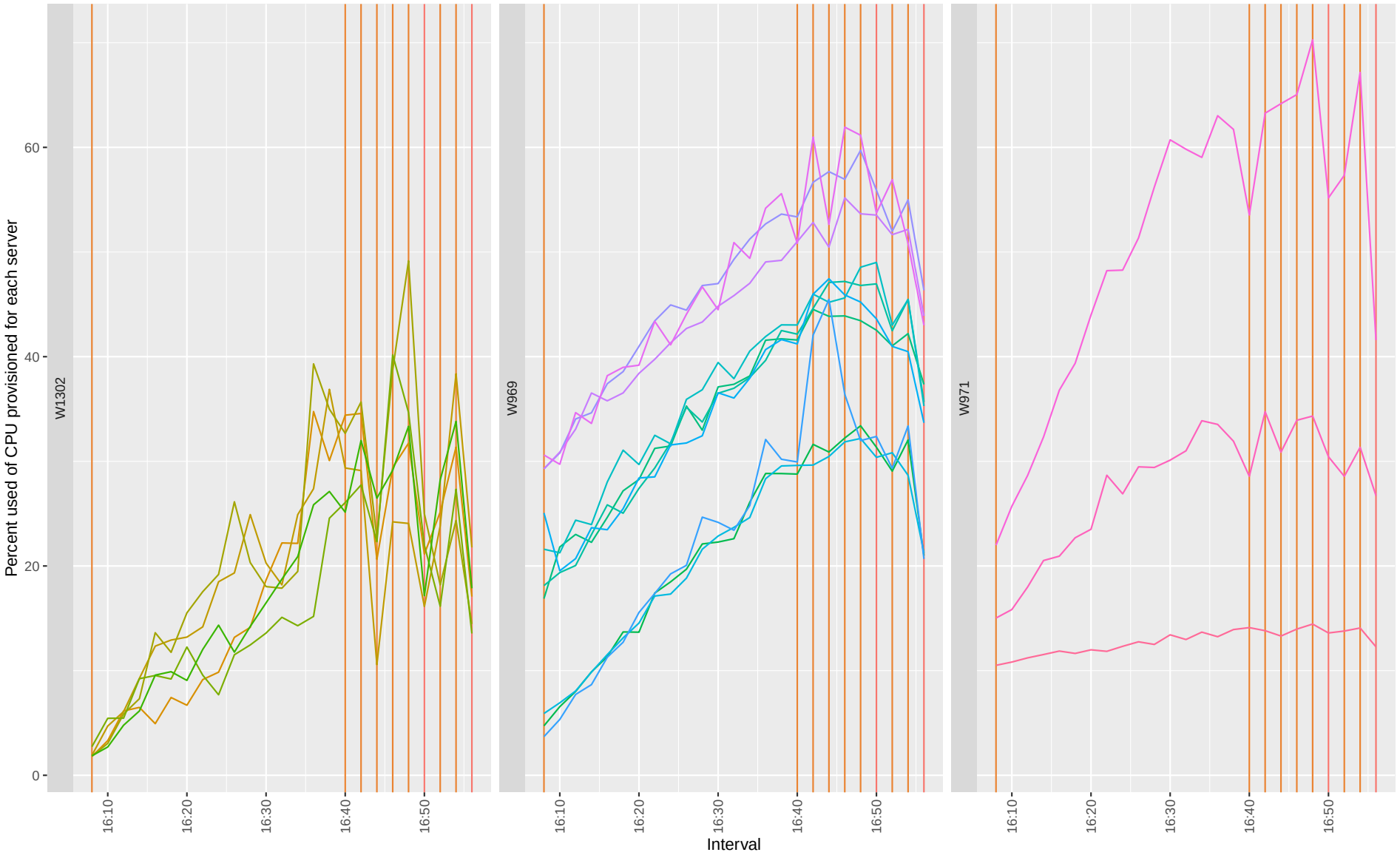
$$t^* = \frac{\widehat{\beta}_1 - k}{\mathrm{SE}(\widehat{\beta}_1)} \quad \text{for testing network bandwidth usage}$$

with the same degrees of freedom as the respective t-statistic on the regression slope where the cluster resource usage is regressed onto each of the cluster members.

CPU usage balance test results

Balance test results in terms of CPU resource consumption for clusters W969, W971, and W1302. The table shows that for these clusters, loads on servers W13021, W13023, W9696, W9698, W97101, and W97103 are significantly out of balance at the $\alpha = 0.01$ level.

Cluster	Server	Servers	Int Est	Int StdErr	Int t-val	Int p-val	Slope Est	Slope StdErr	Slope=1 t-val	Slope=1 p-val	R2	F-stat	df2	p-val
W1302	W13021	5	4.39	1.09	4.01	0.00	0.76	0.07	-3.72	0.00	0.91	134.46	13	0
W1302	W13022	5	-0.07	1.27	-0.06	0.96	0.86	0.07	-2.20	0.05	0.93	171.98	13	0
W1302	W13023	5	1.82	1.44	1.27	0.23	0.73	0.07	-3.84	0.00	0.89	107.17	13	0
W1302	W13024	5	-1.94	2.43	-0.80	0.44	1.45	0.19	2.30	0.04	0.81	55.64	13	0
W1302	W13025	5	0.89	0.90	0.99	0.34	1.04	0.06	0.61	0.55	0.96	310.40	13	0
W969	W9691	10	13.42	0.44	30.67	0.00	0.97	0.02	-1.45	0.17	0.99	1818.81	13	0
W969	W96910	10	-1.33	1.31	-1.01	0.33	1.02	0.04	0.40	0.69	0.98	623.15	13	0
W969	W9692	10	1.17	1.19	0.98	0.34	0.96	0.04	-1.15	0.27	0.98	655.74	13	0
W969	W9693	10	-2.85	1.53	-1.86	0.09	1.01	0.05	0.29	0.78	0.97	503.15	13	0
W969	W9694	10	13.28	0.36	37.17	0.00	0.98	0.02	-1.06	0.31	1.00	2764.69	13	0
W969	W9695	10	0.64	1.11	0.58	0.57	0.99	0.04	-0.38	0.71	0.98	780.83	13	0
W969	W9696	10	15.07	0.60	25.13	0.00	0.85	0.03	-5.14	0.00	0.98	813.53	13	0
W969	W9697	10	-12.05	1.34	-9.01	0.00	0.99	0.03	-0.37	0.72	0.99	1051.61	13	0
W969	W9698	10	-19.43	1.75	-11.08	0.00	1.23	0.04	5.34	0.00	0.98	835.59	13	0
W969	W9699	10	-8.53	2.30	-3.71	0.00	0.92	0.05	-1.60	0.13	0.96	300.40	13	0
W971	W97101	3	5.32	0.43	12.52	0.00	0.45	0.01	-63.66	0.00	1.00	2710.43	13	0
W971	W97102	3	0.88	1.15	0.77	0.45	0.98	0.04	-0.48	0.64	0.98	531.12	13	0
W971	W97103	3	-45.03	6.91	-6.52	0.00	5.80	0.56	8.62	0.00	0.89	108.53	13	0

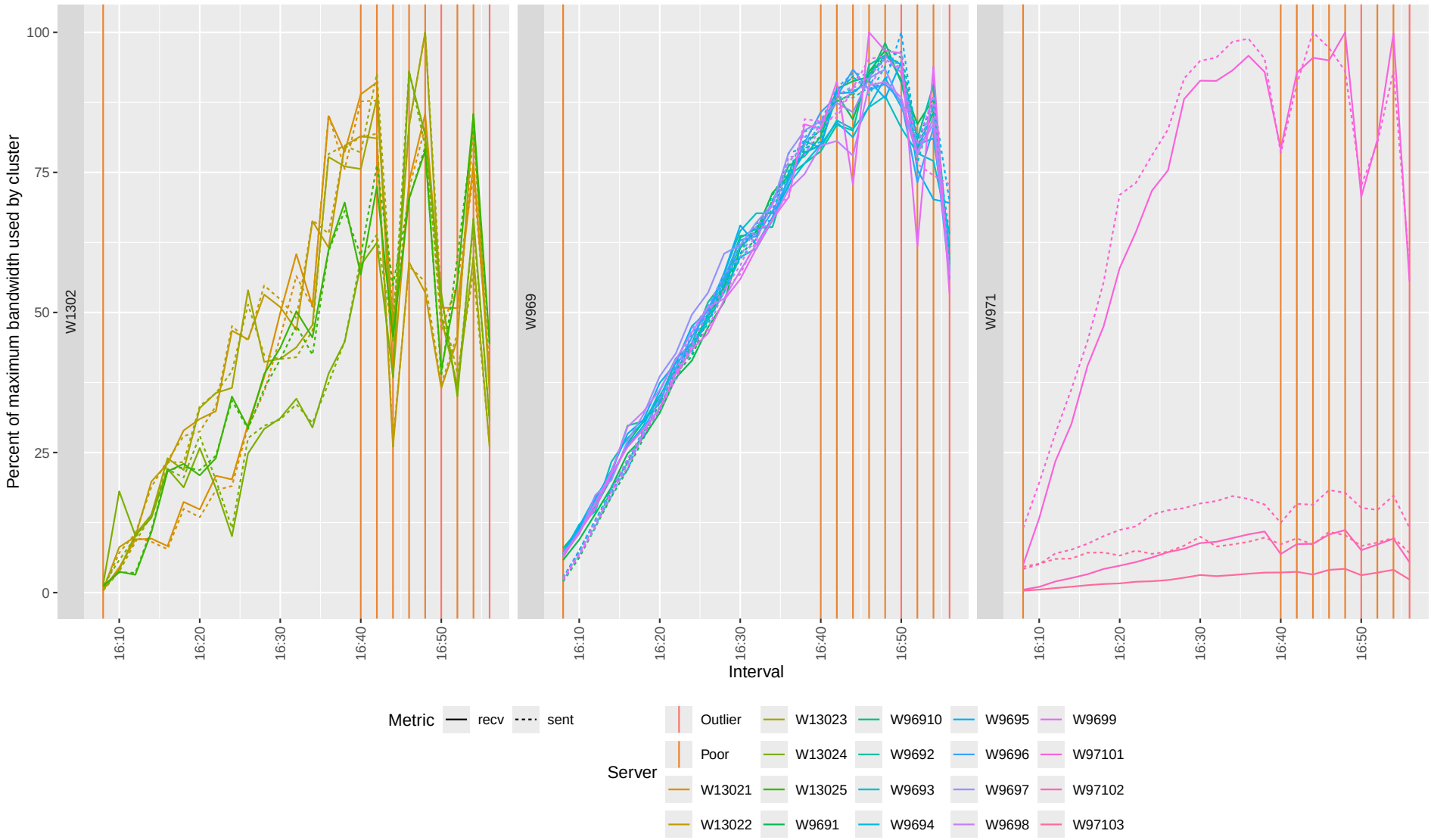

Percent used of CPU provisioned for each server
0
20
40
60
W1302
W969
W971
16:10
16:20
16:30
16:40
16:50
Interval

Network usage balance test results

Balance test results in terms of network resource consumption for clusters W969, W971, and W1302. The network resource usage on servers W13021, W13021, W13024, W9694, W97101, W97101, W97102, W97102, W97103, and W97103 are significantly out of balance at the $\alpha = 0.01$ level. However, for server W9694 this appears to be a false positive as the slope estimate is only slightly over 10 (at 10.4).

Clstr	Srvr	Srvrs	metric	Int Est	Int StdErr	Int t-val	Int p-val	Slope Est	Slope StdErr	Slope=k t-val	Slope=k p-val	R2	F-stat	df2	p-val
W969	W9691	10	sent	2.16	0.75	2.88	0.01	9.90	0.08	-1.35	0.20	1.00	16768.83	13	0
W969	W9691	10	recv	0.92	0.22	4.12	0.00	9.47	0.18	-2.86	0.01	1.00	2663.21	13	0
W969	W96910	10	sent	0.78	1.06	0.74	0.47	9.90	0.11	-0.94	0.37	1.00	8677.34	13	0
W969	W96910	10	recv	0.21	0.15	1.36	0.20	9.79	0.13	-1.70	0.11	1.00	6121.57	13	0
W969	W9692	10	sent	0.49	1.03	0.47	0.65	9.94	0.10	-0.57	0.58	1.00	9137.36	13	0
W969	W9692	10	recv	-0.15	0.23	-0.64	0.53	10.09	0.19	0.49	0.63	1.00	2855.16	13	0
W969	W9693	10	recv	0.05	0.25	0.19	0.85	9.82	0.20	-0.87	0.40	0.99	2368.68	13	0
W969	W9693	10	sent	-0.09	1.00	-0.09	0.93	9.92	0.10	-0.81	0.43	1.00	9845.29	13	0
W969	W9694	10	recv	-0.33	0.13	-2.46	0.03	10.45	0.11	4.07	0.00	1.00	9014.51	13	0
W969	W9694	10	sent	-1.06	0.99	-1.07	0.30	10.14	0.10	1.39	0.19	1.00	10296.58	13	0
W969	W9695	10	sent	-0.68	1.69	-0.40	0.69	9.99	0.17	-0.08	0.94	1.00	3481.96	13	0
W969	W9695	10	recv	-0.34	0.22	-1.51	0.16	10.11	0.18	0.64	0.54	1.00	3172.30	13	0
W969	W9696	10	recv	0.17	0.10	1.67	0.12	9.82	0.08	-2.12	0.05	1.00	13506.59	13	0
W969	W9696	10	sent	-0.39	0.83	-0.47	0.64	9.97	0.08	-0.32	0.76	1.00	14492.47	13	0
W969	W9697	10	recv	0.09	0.25	0.35	0.73	9.46	0.19	-2.82	0.01	0.99	2462.00	13	0
W969	W9697	10	sent	-0.14	1.19	-0.12	0.91	9.88	0.12	-0.99	0.34	1.00	6953.87	13	0
W969	W9698	10	recv	-0.63	0.27	-2.31	0.04	10.67	0.23	2.96	0.01	0.99	2209.59	13	0
W969	W9698	10	sent	-0.92	1.50	-0.61	0.55	10.16	0.15	1.03	0.32	1.00	4417.65	13	0
W969	W9699	10	sent	1.82	2.31	0.79	0.44	9.98	0.24	-0.07	0.95	0.99	1775.32	13	0
W969	W9699	10	recv	0.40	0.35	1.14	0.28	9.98	0.29	-0.06	0.95	0.99	1171.69	13	0

Clstr	Srvr	Srvrs	metric	Int Est	Int StdErr	Int t-val	Int p-val	Slope Est	Slope StdErr	Slope=k t-val	Slope=k p-val	R2	F-stat	df2	p-val
W1302	W13021	5	recv	23.17	4.53	5.11	0.00	3.49	0.24	-6.22	0.00	0.94	207.86	13	0
W1302	W13021	5	sent	9.32	2.01	4.65	0.00	3.56	0.28	-5.12	0.00	0.92	159.85	13	0
W1302	W13022	5	recv	-3.49	7.43	-0.47	0.65	4.36	0.37	-1.74	0.10	0.91	139.48	13	0
W1302	W13022	5	sent	-1.66	2.66	-0.62	0.54	4.31	0.33	-2.08	0.06	0.93	170.08	13	0
W1302	W13023	5	recv	3.84	6.39	0.60	0.56	4.25	0.33	-2.25	0.04	0.93	162.06	13	0
W1302	W13023	5	sent	1.29	2.25	0.57	0.58	4.21	0.29	-2.68	0.02	0.94	204.34	13	0
W1302	W13024	5	sent	-5.14	4.55	-1.13	0.28	7.98	0.97	3.07	0.01	0.84	67.82	13	0
W1302	W13024	5	recv	-16.50	13.44	-1.23	0.24	8.28	1.13	2.91	0.01	0.81	53.95	13	0
W1302	W13025	5	sent	1.79	1.43	1.25	0.23	4.97	0.22	-0.15	0.88	0.97	494.06	13	0
W1302	W13025	5	recv	7.19	3.77	1.91	0.08	4.73	0.23	-1.21	0.25	0.97	438.33	13	0
W971	W97101	3	recv	-1.50	0.88	-1.71	0.11	1.14	0.01	-217.29	0.00	1.00	17961.45	13	0
W971	W97101	3	sent	1.77	0.19	9.54	0.00	1.19	0.01	-194.93	0.00	1.00	16349.78	13	0
W971	W97102	3	recv	16.64	5.17	3.22	0.01	9.94	0.51	13.58	0.00	0.97	378.27	13	0
W971	W97102	3	sent	-2.71	1.00	-2.71	0.02	8.11	0.29	17.67	0.00	0.98	786.41	13	0
W971	W97103	3	recv	6.22	6.19	1.00	0.33	32.49	1.83	16.12	0.00	0.96	315.36	13	0
W971	W97103	3	sent	-16.91	6.24	-2.71	0.02	20.41	3.06	5.69	0.00	0.77	44.49	13	0


Percent of maximum bandwidth used by cluster
100
75
50
25
0
W1302
W969
W971
16:10
16:20
16:30
16:40
16:50
Interval
Metric
recv
sent
Server
Outlier
Poor
W13021
W13022
W13023
W13024
W13025
W9691
W96910
W9692
W9693
W9694
W9695
W9696
W9697
W9698
W9699
W97101
W97102
W97103

Conclusion

Conclusion

- Supporting informed decisions with compact common scale graphics of performance and resource usage.
- Execution of tests as planned repeatable experiments.
- Estimate performance (using local regressions) under load conditions that will be important in the field.
- Resolve resource usage into background load and customer activity load (using linear regressions).
- Track performance and resource usage over repeated experiments, and detect significant changes when compared to history (using the bootstrap test).
- Model customer driving activity against system-under-test (I/O FSM simulation existing).
- Model third-party behaviour (finite mixture models of response times).

Ongoing work:

- Consider changes required so that response times are not just point estimates, but include sufficient detail to test SLAs
- Type I error - tuning significance levels. A Type I error is much better than an outage in the field.
- Narrow confidence intervals - sensitive to changes that may not be material.
- Tidy up reporting and pipeline integration and deploy to field as a pilot.
- Pipeline integration

Thank you