Table of Contents

Talos Signal From Noise Mockups

This document presents a description of a user interface framework that would facilitate rapid exploration of different data types using multiple visualization strategies. The document does not constitute a functional specification; it's more like a collection of ideas to discuss. If we decide to pursue some of them, we could pick a subset and target a Q1 deliverable.

The primary purpose of the user interface presented is to address some of the observations described in https://wiki.mozilla.org/Metrics/Talos_Investigation and to provide an application that will support an iterative analysis workflow with a flexible user interface.

One of the core ideas presented is to allow for multiple visualization strategies for the same data set in an effort to address a wider array of use cases without having lots of disconnected web pages. The UI presented will scale to a large number of diverse data sets, visualizations, and use cases.

User Type Definitions

There are four types of potential users to consider:

NOTE: *A changeset is a set of changes in a source code repository that have been pushed to Mozilla-central and have a set of associated test results. A changeset might include many modified source code files and may have multiple authors but should have a 1 to 1 relationship with a complete set of TBPL test results*

Sheriff

The Sheriff is assigned to monitor changesets to a particular source tree using https://tbpl.mozilla.org/ . The sheriff is trying to protect the source tree from regressions or an assault of problematic patches that would require shutting down a source tree and possibly backing out changesets.

Firefox Developer

Pushes a changeset to mozilla-central. Looking to see if their patch or set of patches has passed all tests on a particular source branch on all platforms. If a test fails or a build error occurs the developer will need to determine what caused the problem in the changeset supplied.

Talos Developer

Adds a new feature to a Talos performance test with an expected change in numbers. The new test will need to be run side-by-side with the original for a ~week to see if the change introduces a regression, some other undesired outcome, or the target outcome.

Investigator

Could be looking for a number of things. Perhaps there has been some performance degradation over time with an unknown cause. The investigator will have the most demanding use cases and will likely require an analysis workflow rather than a hardcoded presentation of one data type.

Manager

A manager would want to keep track of a developer or group of developer's changesets and how they perform.

Data Presentation Goals

Goal 1

Create a visual display that enables the comparison of more Talos tests across relevant source trees and platforms.

Goal 2

Create data visualizations that provide improved support for trend observations in Talos test results over time or multiple changesets.

Goal 3

Provide facilities for drilling down to a much lower level of detail for any given set of changesets or time ranges.

Goal 4

Display performance results for the set of web pages associated with a given Talos test type. Allow a user to compare pageset test results across platforms and different changesets.

Goal 5

Allow for users to add persistent comments to data points so that user observations can be easily shared and the same analysis does not have to be repeated.

User Interface Goals

Goal 1

Define a generic representation of data, called a “data view”. A webpage can contain an unlimited number of data views. Data views can be connected across multiple web pages. This concept is also illustrated in the bughunter wiki page.

Goal 2

Extend the existing user interface to manage multiple disparate forms of data by abstracting navigation, application controls, and data visualization types into a set of widgets that maximize the area for data content display.

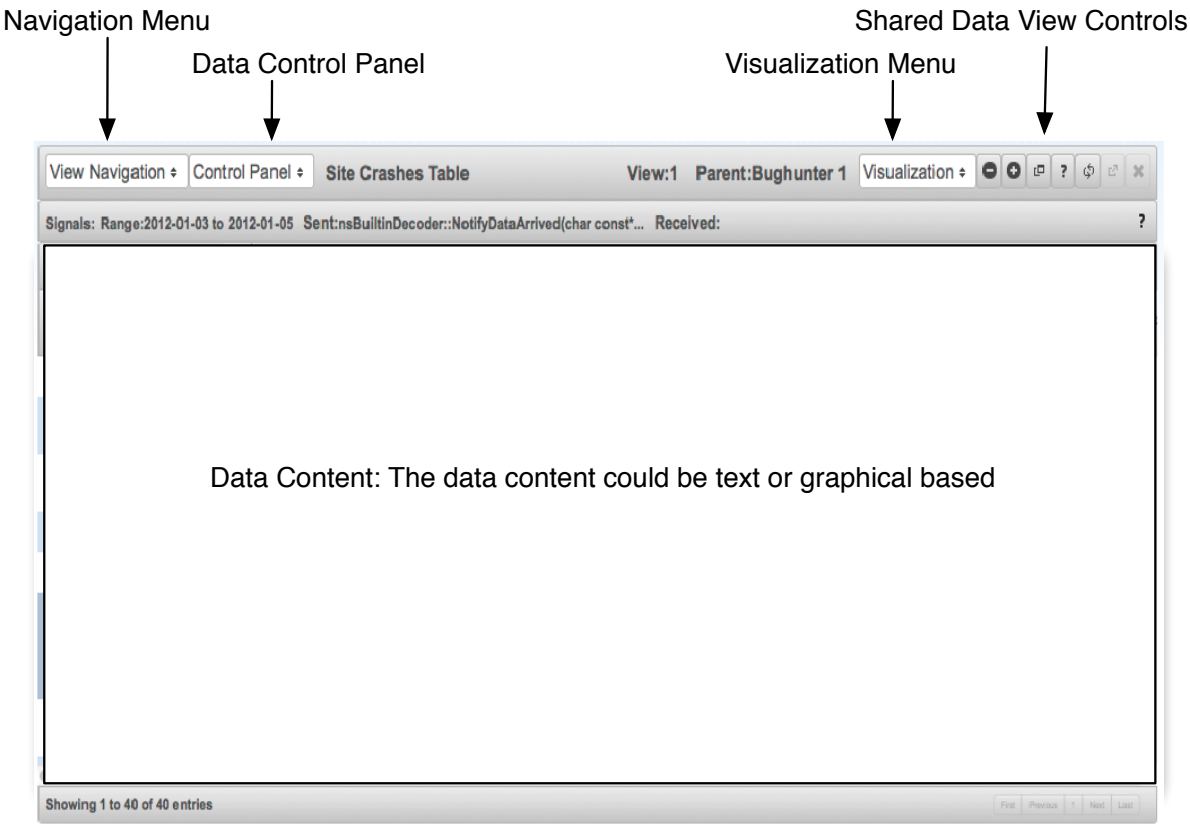
Goal 3

Maintain context between different data types by allowing one data view to send “signals” to another. Essentially, use DOM events to initialize data views within a single page or a set of connected pages rather than opening a disconnected page every time a user selects a link.

Goal 4

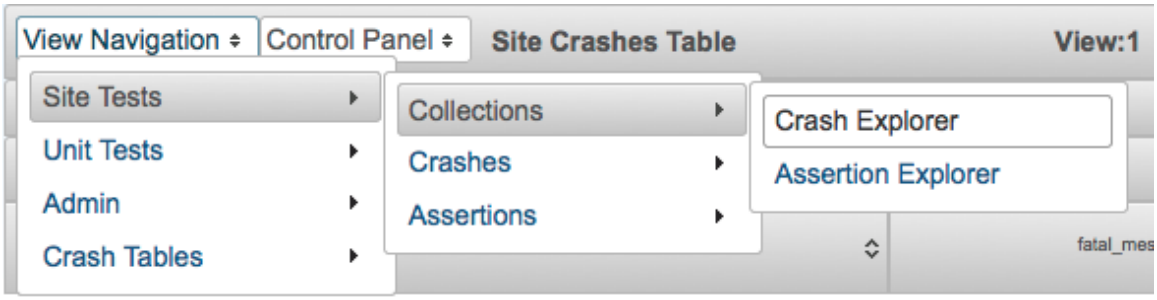
Create a user interface that can be rapidly extended with new data types and visualization approaches without having to build additional websites.

Data View Example



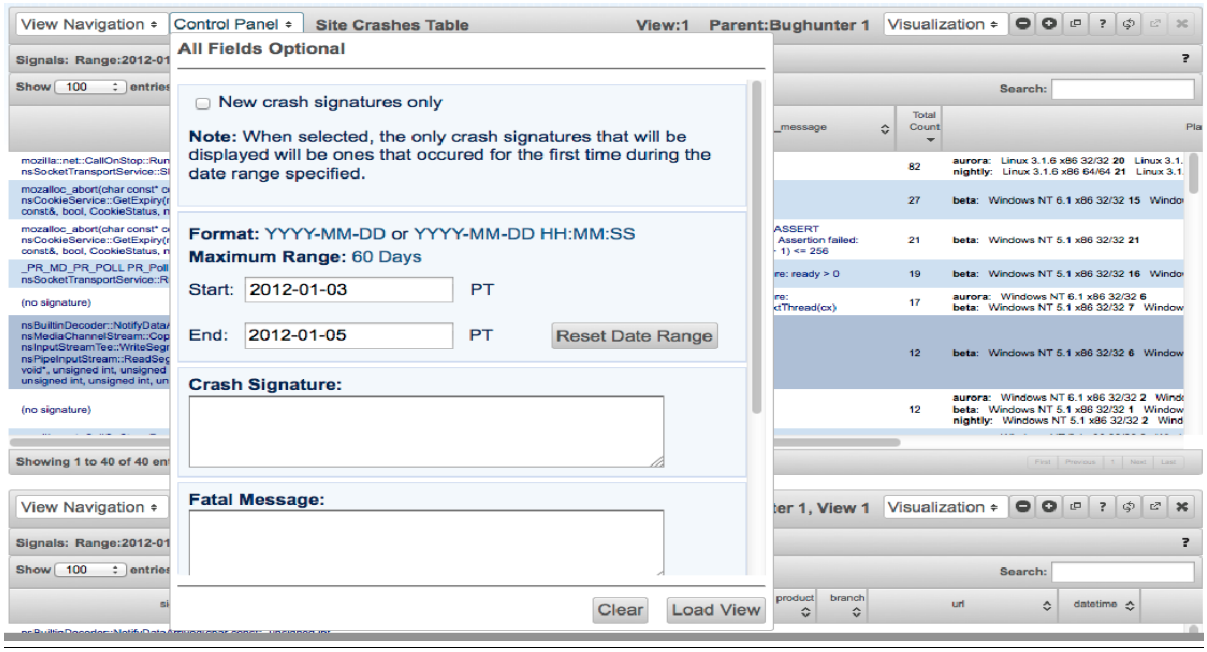
An empty data view is presented above. All data views would contain this basic layout.

Navigation Menu



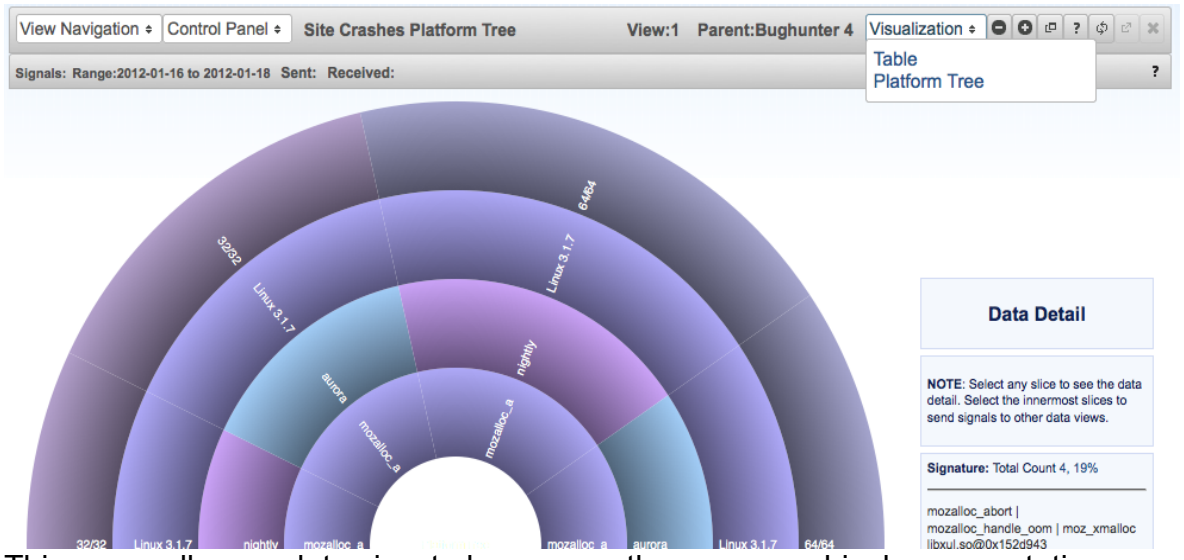
The navigation menu allows the user to change from one data view to another. This could enable the user interface to display Talos summary data, Talos test result details, changeset summaries, pageset data associated with a particular set of test results etc… Basically we could scale to a lot of different data views.

Data Control Panel



This is a dropdown panel that allows any set of controls associated with a given data view to be accessed without using the content display area. The controls could be anything: date range specifiers, textareas, checkboxes, buttons etc…

Visualization Menu



This menu allows a data view to have more than one graphical representation. In the case of the screen above, the representations available are Table and Platform Tree but could be an unlimited number of possibilities.

Shared Data View Controls

This button set provides access to common functionality shared across all views. From left to right:

Decrease View Panel Size: Decrease the display size of a view's data representation.

Increase View Panel Size: Increase the display size of a view's data representation.

Open New View: Opens a modal window that gives users the option to open another data view.

Help: Application help.

Refresh: Reload the data from its original source.

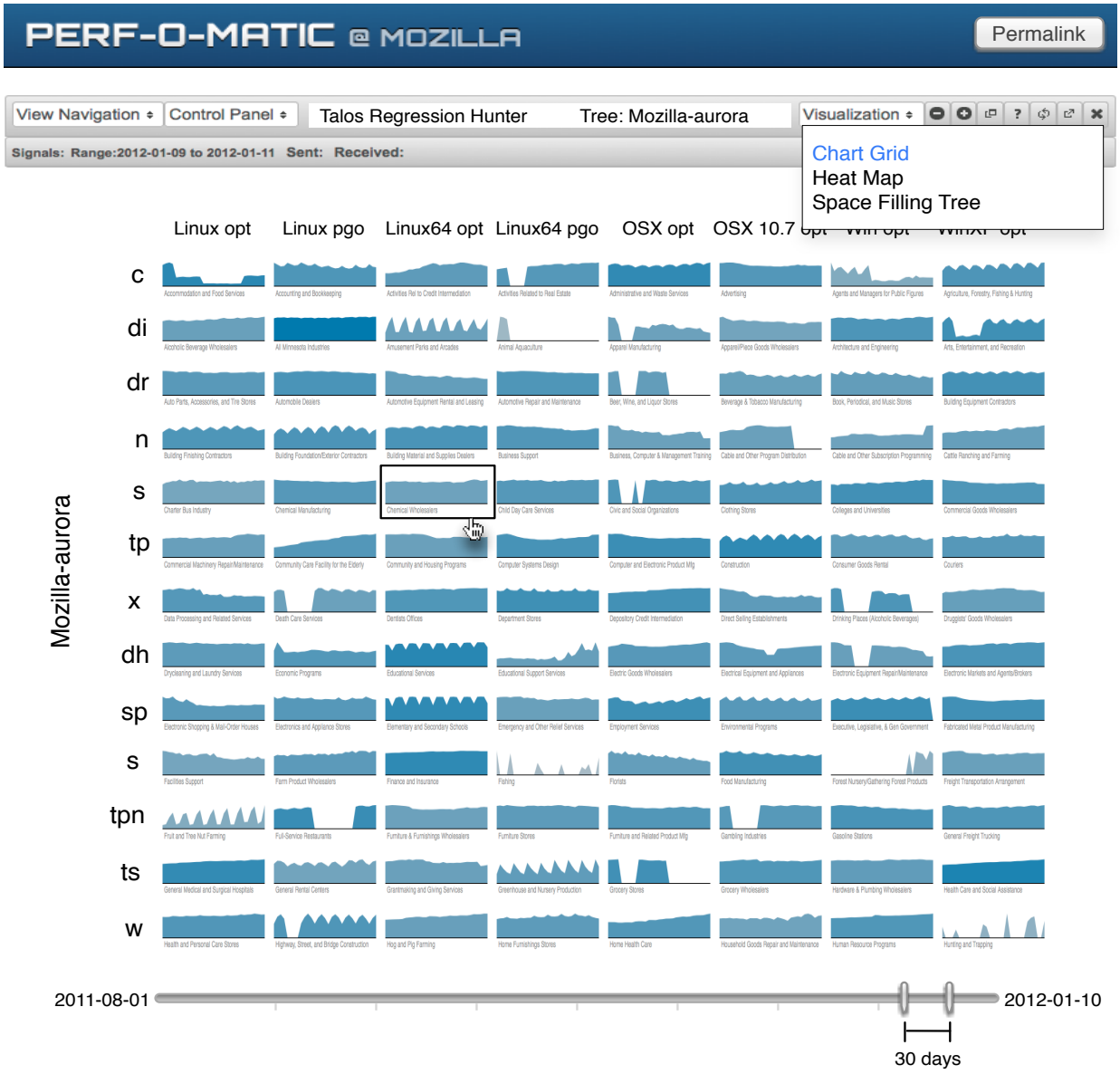
Move To New Window: Open the data view in a new browser tab or window.

Close: Close the data view

Signals

Data views can send signals to one another. A signal is a DOM event with a data payload that enables a listener to initialize to a shared type of data. The mock displayed below shows three data views, all displaying tabular data. When a user selects a cell containing a link an event is fired that the other data views listen for, the payload of the event allows the child data views to initialize to the data the user selected. This allows the user to maintain context, if a new page is opened for every link context is lost, it's also harder to rapidly browse multiple connected data types. This same concept works across multiple browser windows when the visual capacity of a single page is exhausted. Events can be passed between multiple browser tabs/pages to get the most out of a display.

Talos Regression Hunter: Chart Grid



Users

Firefox Developer, Talos Developer, Sheriff and Investigator

Entry Points

Coming in from a particular source tree displayed in TBPL.

Use Cases

1. User wants to determine what platform trends can be observed over a given time range for a particular test type.
2. A source tree is showing repeated problems with the same Talos test, user wants to determine how long this has been occurring for.

3. A source tree has just started to show test failure, user wants to determine if this is consistent with recent results.
4. Changes were made to a Talos test, user wants to determine how do the new results compare.
5. A changeset was pushed to Mozilla-central and has unexpected results on a particular platform Talos test combination. User wants to determine how do other test results compare.

Data View Description

The mock displayed above takes the same concept found on the perf-o-matic landing page and puts the charts on the same x-y axis to save space. The charts would each plot the same Talos data that they are currently displaying but for the date range specified on the date range slider.

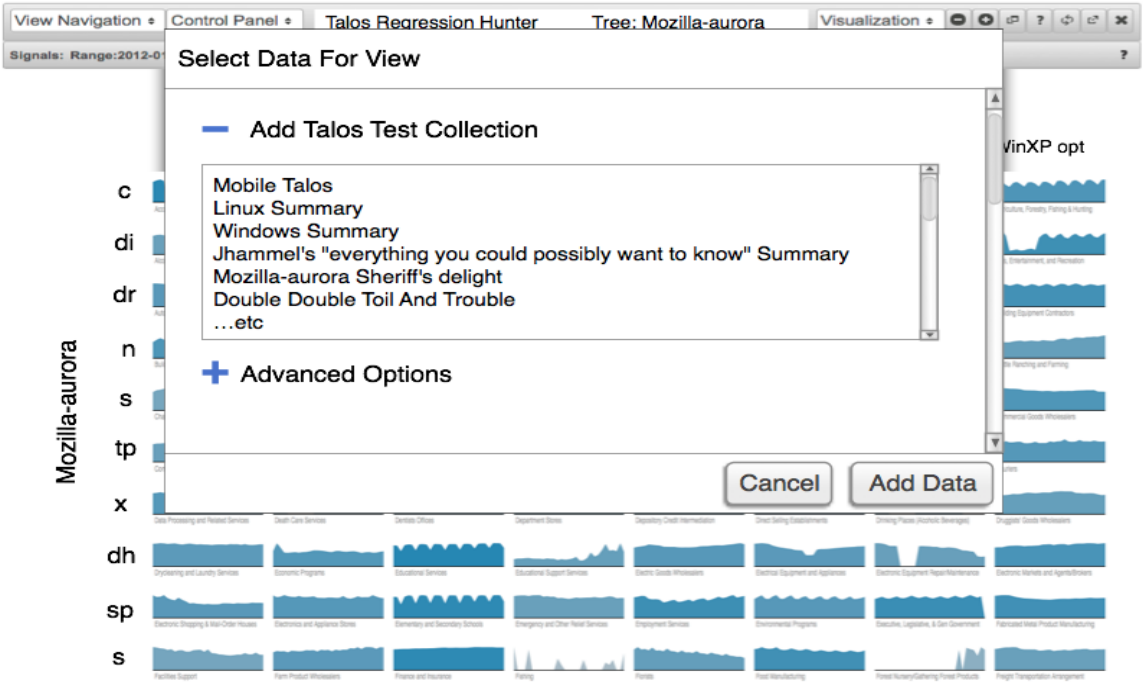
The date range slider on the content pane controls the date range of the data examined in the charts. This may not be possible given the quantity of data but if it were possible, it would allow the user to quickly see changes over different time ranges. If the slider is not possible we could use a date picker strategy but embed the picker on the dropdown control panel because there would be no real-time drag response.

The chart thumbnails would be selectable. When selected they would send a signal to other child data views containing the date range, source tree, Talos test, and platform.

We could also add an orange guideline to each thumb chart indicating where the standard cutoff lies to give an idea of where the mean is drifting above it.

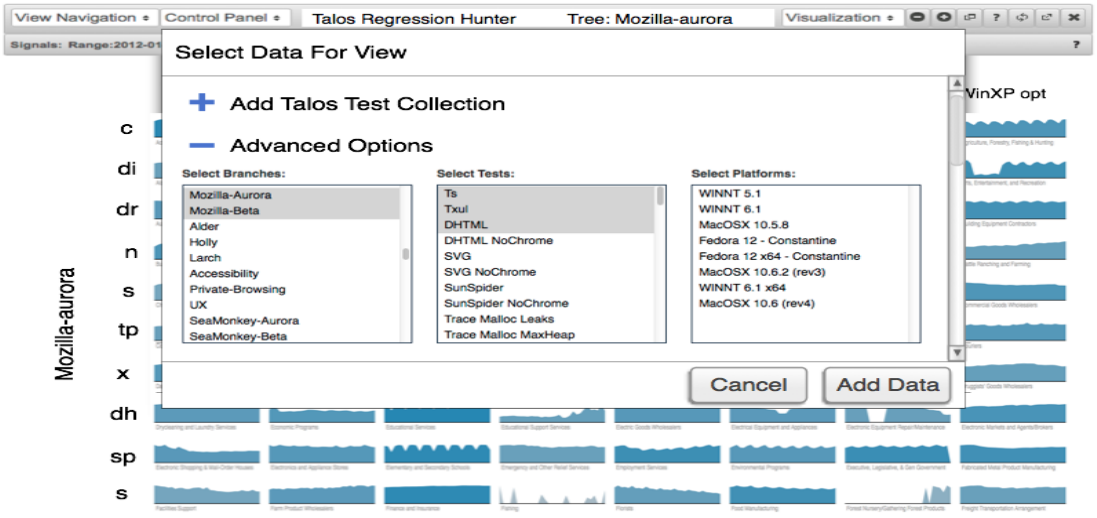
Another possible dimension for information encoding would be the color intensity or color of the graph thumbs. We could keep consistent with the test color encodings on TBPL by running a two color ramp from green to orange, where the chart with the lowest mean (mean of means of medians… or something like that) under the allowed range would be the most green, as values approach the cutoff of the allowed range they would become more orange, and any mean exceeding the range would be completely orange.

The dropdown control panel would be a consolidation of the existing controls but would include separate options for the 80% use case and advanced options.

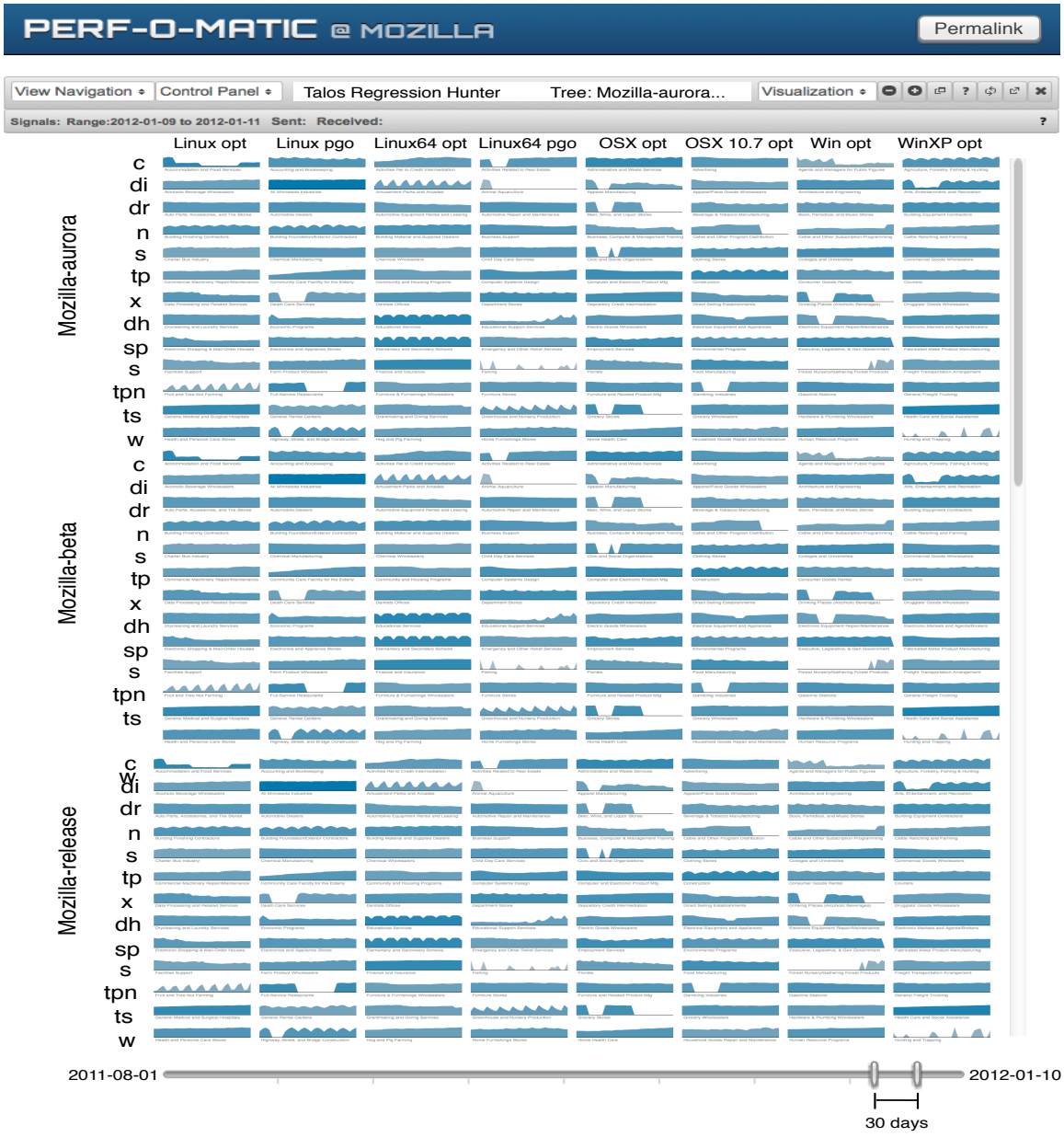


The 80% use cases would be pre-defined combinations of source tree, Talos test type, and platform. These combinations would be given intuitive names and descriptions that would help the user make the appropriate choice for what they are looking for. This would help non-expert users get straight to the combination they need to analyze.

The existing perf-o-matic menu controls would be present under advanced options.



This graphical presentation could be extended to include multiple source trees by vertically stacking source trees with a selection of Talos tests to investigate. This would work best with a small number of Talos tests. This would enable comparisons being made across source trees.



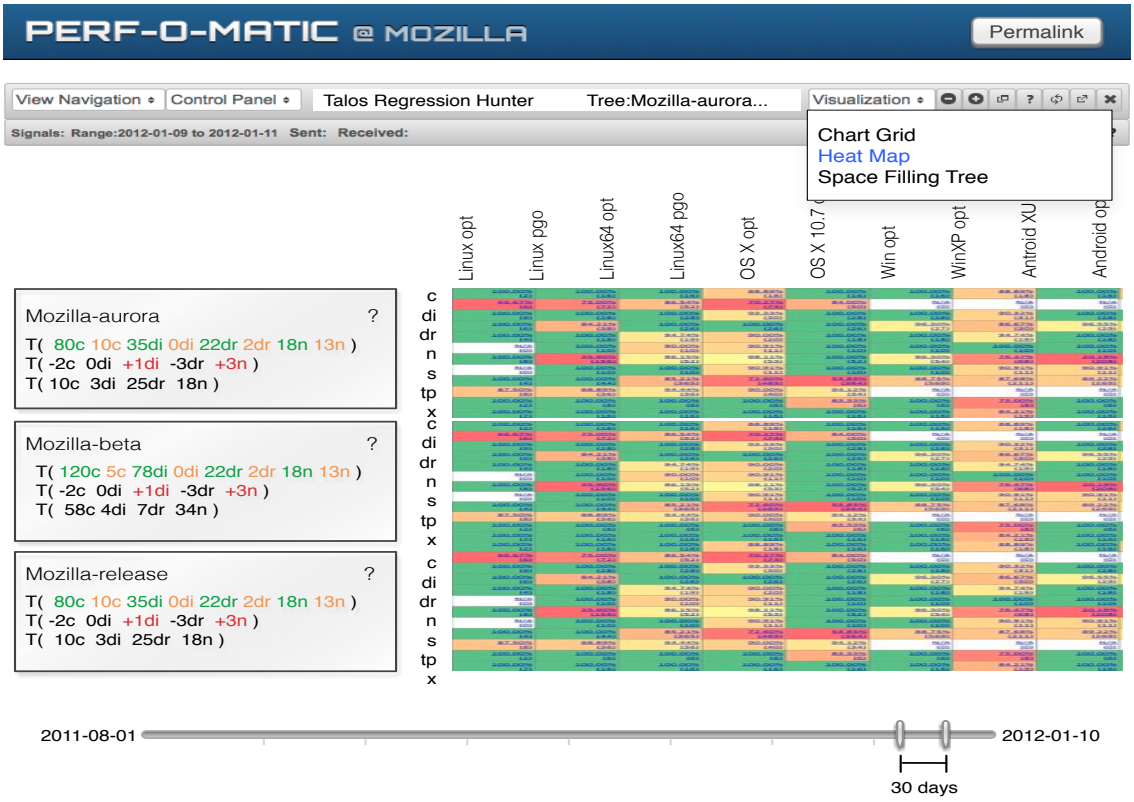
Talos Regression Hunter: Heat Map

Another possible visualization that would help assess trends in Talos test data is a heat map. A user could toggle between the Chart Grid and Heat Map visualizations by selecting options in the Visualization menu. In the heat map depiction below, Talos tests are laid out on each row while each column specifies

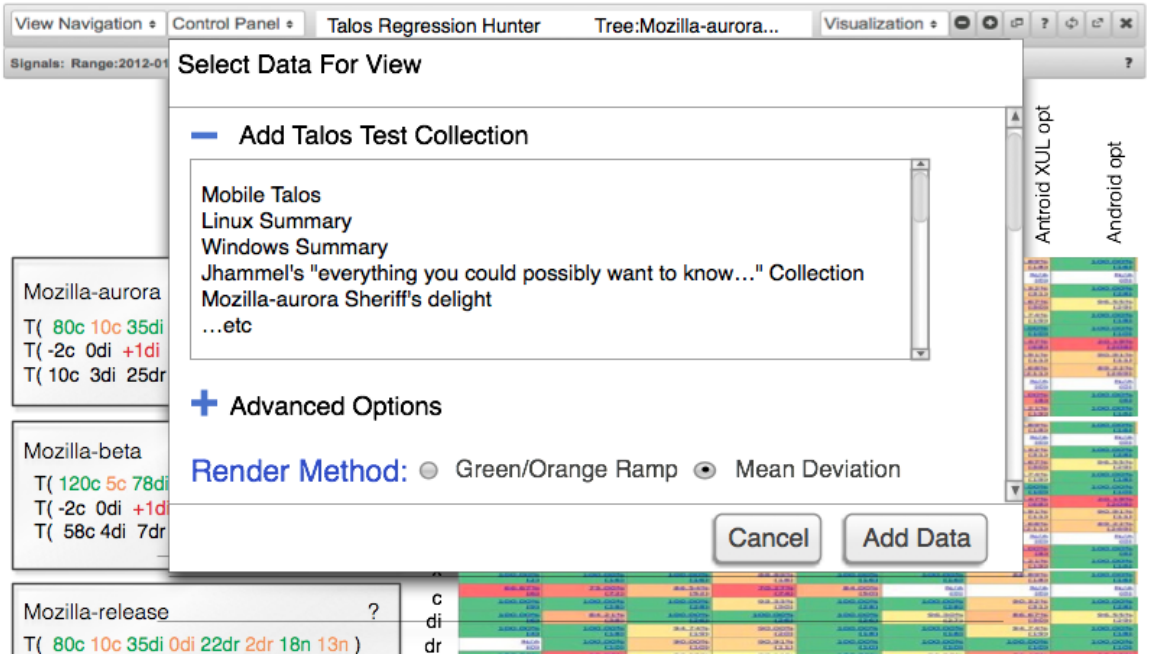
a different platform. The gray boxes on the left indicate source tree type and anchor a set of Talos tests to them, allowing for the comparison of multiple source trees or just a single source tree. Each cell would represent a mean of means of median values that could be compared to the cutoff, indicating test failure, to determine what color should be assigned.

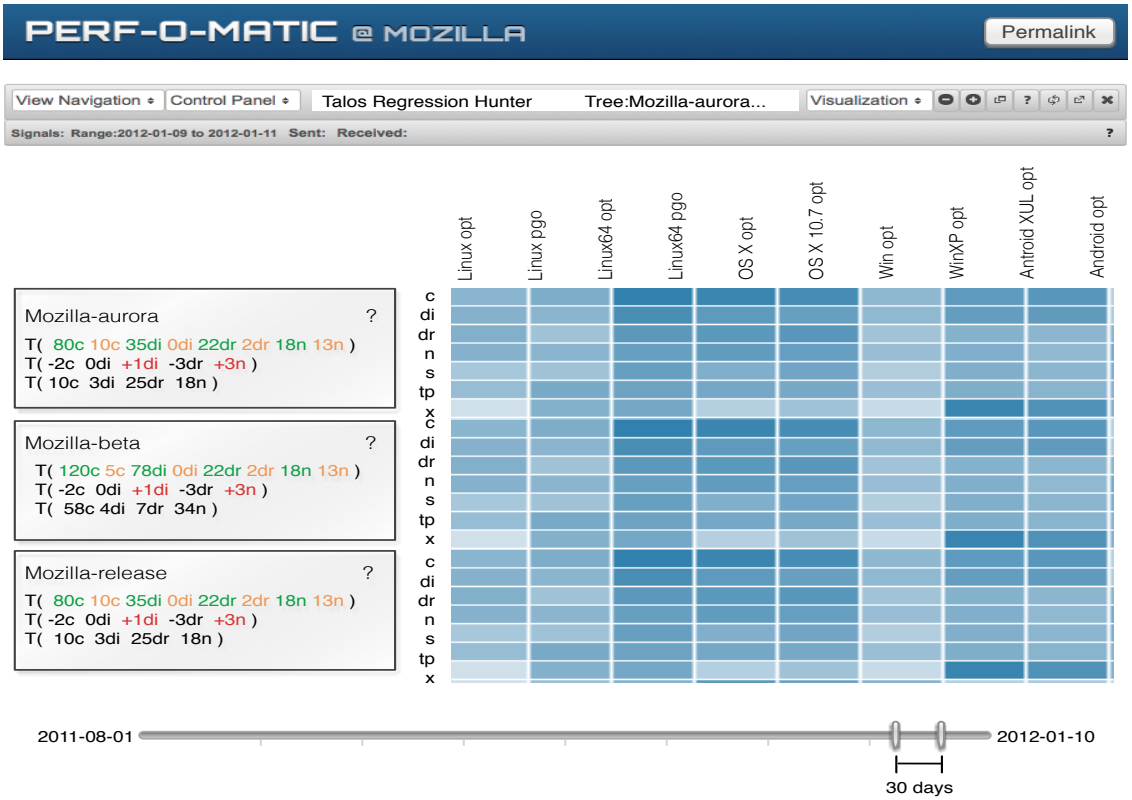
The color ramp would be run from green to orange or dark orange if a value dramatically exceeds the cutoff, keeping consistent with the TBPL colors. The mock does not run a real color ramp but it was the closest thing I could find that I could easily scrape, use your imagination, as the mean value approaches the cutoff it would move from green to orange and if it surpasses the cutoff it could move to dark orange or red.

Each source tree box on the left could contain additional details (from top to bottom), such as how many Talos tests were within the cutoff or over, the average deviation from the cutoff, or deviation from the mean to give some indication of variation.



The heat map could be colored using more than one method. The control panel depicted in the mock below shows a radio button providing the option to color by Mean Deviation. When selected the color scheme could look something like this.



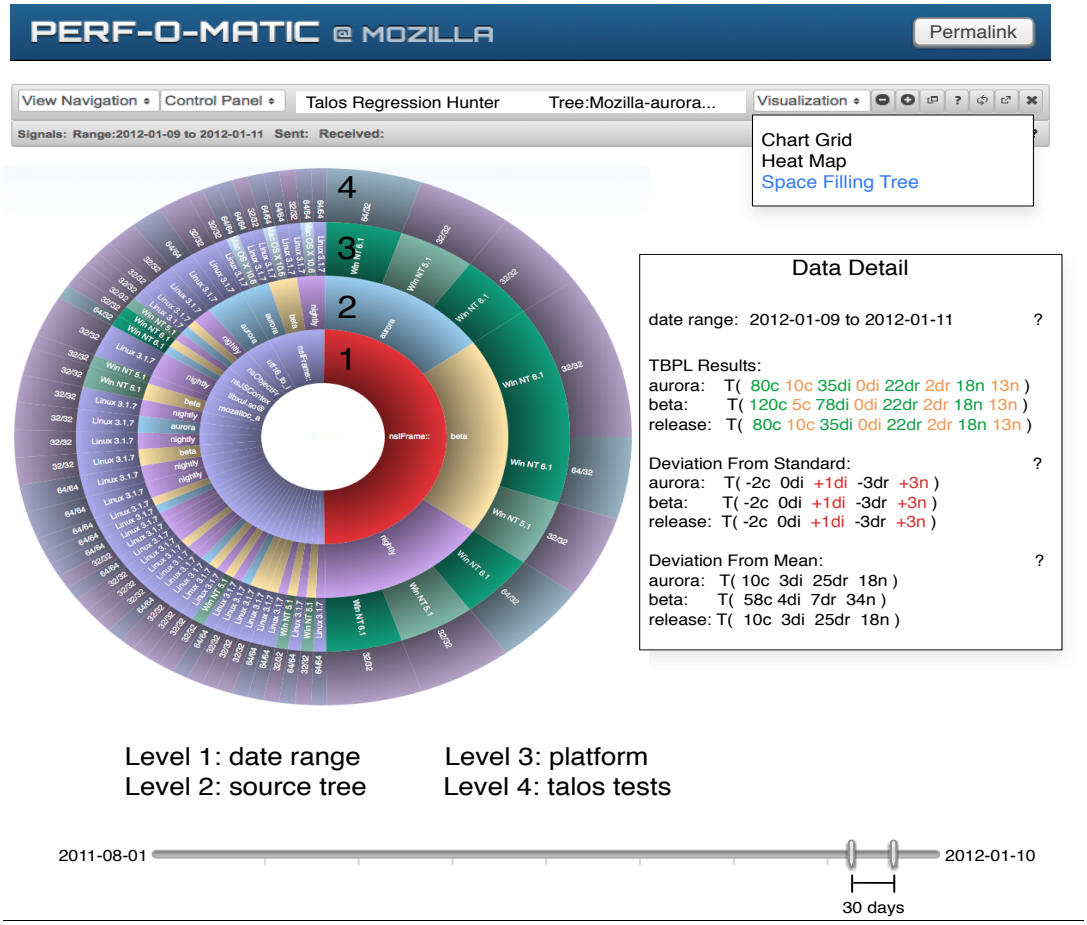


Coloring by Mean Deviation would give a qualitative assessment of which source tree, Talos test, and platform combinations exhibited the most variation over the specified time range.

When a cell in the heat map is selected it would behave the same as a chart thumb on the Chart Grid and would send a signal to other child data views containing the associated date range, source tree, Talos test, and platform.

Talos Regression Hunter: Space Filling Tree

A space-filling tree is an adjacency diagram where rather than drawing a link between the parent/child hierarchies, nodes are drawn as solid areas and their placement indicates their position in the hierarchy. The space filling characteristics of the nodes make them ideal for encoding relative size differences in a hierarchy.



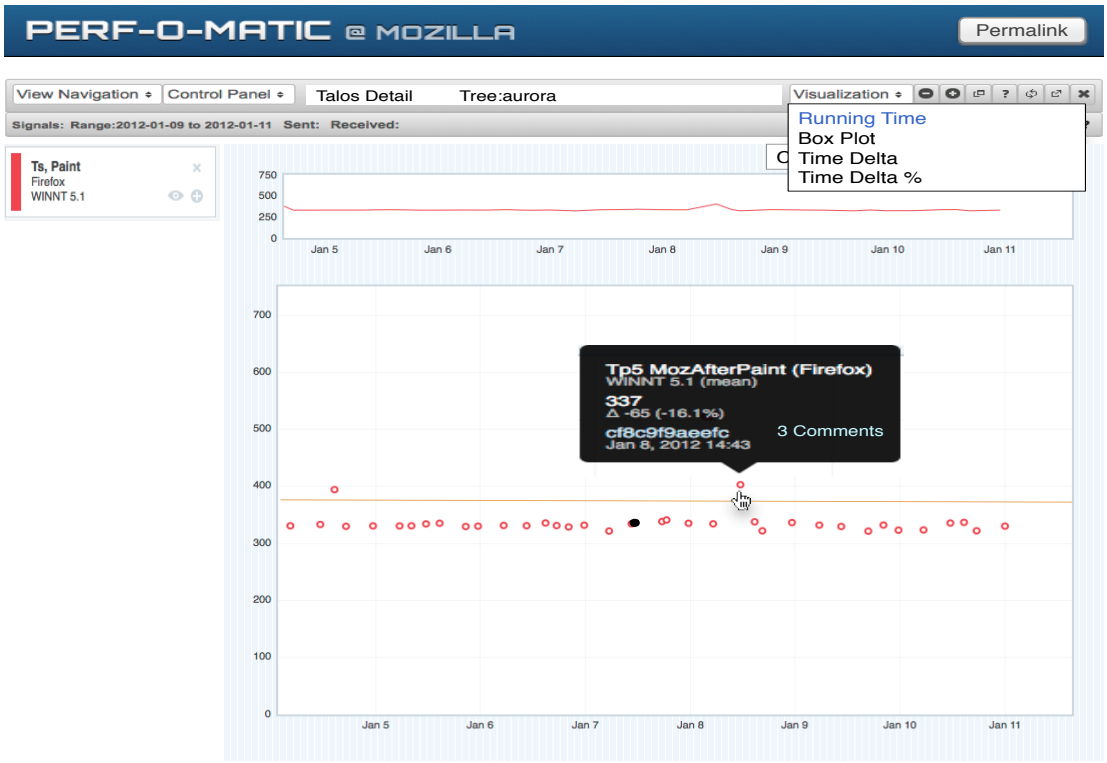
The first level of the tree would represent the date range specified with the slider. Time granularity for date ranges would need to be chosen to prevent too many branches being added to the tree. Too many branches would reduce the utility of the size comparison capacity of the chart. So if the date range was 30 days and the granularity was 10 days, there would be 3 nodes in level one that span the 30-day range. The second level of the tree would specify the source tree to be examined. The third level would contain platform information. The fourth level would contain the Talos tests to be examined.

The node size of the Talos tests in the fourth level would correspond to the mean value, the larger the node size the closer to the cutoff, and we could again use a green to orange color ramp to indicate where the cutoff fits in. As you go up the tree the node sizes would be proportional to the sum of all of the contained tests.

Here we could color code for easy recognition of source tree and platform information. The screen shot shown contains real firefox crash data associated with the source trees: aurora, beta, and nightly. The crash data is not relevant to the Talos test data but the use of the visualization technique and the hierarchical nature of source tree type combined with platform data are.

When a node in the tree is selected it would behave the same as a chart thumb on the Chart Grid and would send a signal to other child data views containing the associated date range, source tree, Talos test, and platform. In addition to that it could display a detail panel to the right that would contain relevant text based data to the tree branch selected.

Talos Detail: Running Time



Users: Firefox Developer, Talos Developer, Sheriff and Investigator

Entry Points: Application entry occurs by selecting a chart or graphical element in Talos Regression Hunter that sends a signal containing a date range, source tree, Talos test, and platform specifier. A user could select multiple Talos Regression Hunter elements to load more than one dataset.

Use Cases:

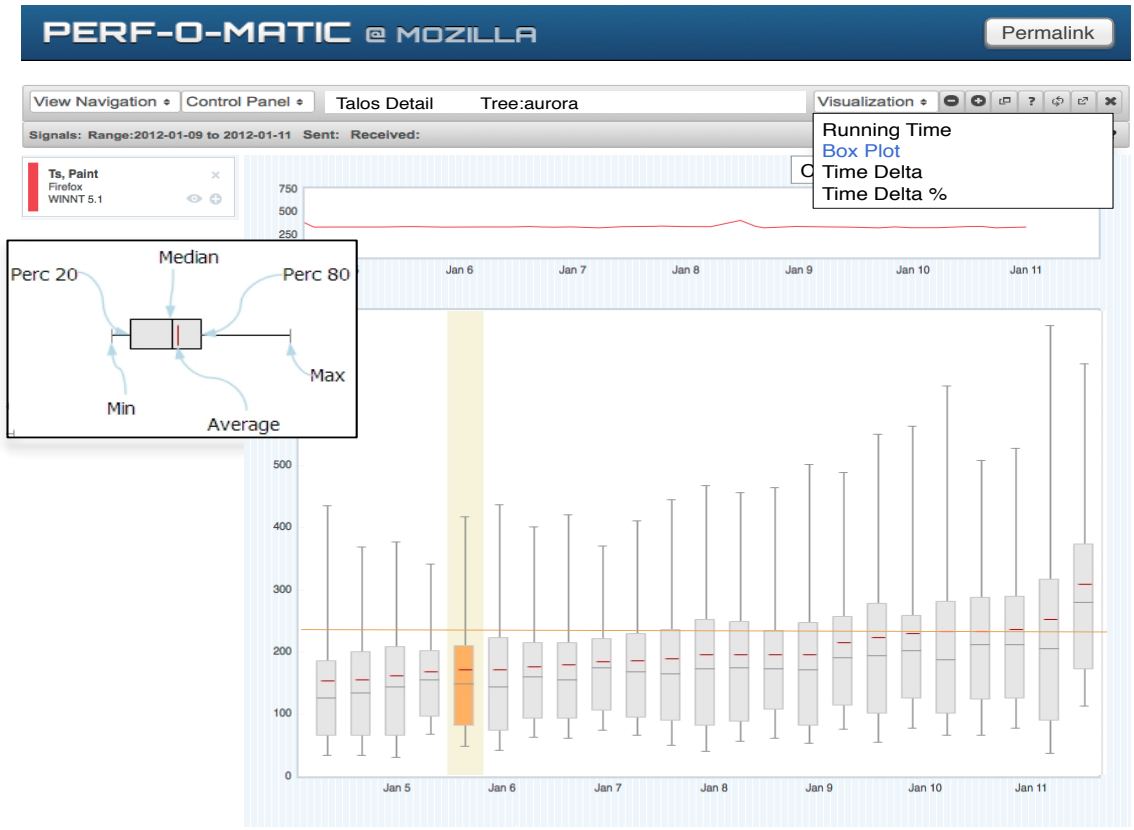
1. User makes an observation using Talos Regression hunter and wants to explore the data further.
2. Same use cases as Talos Regression hunter but for a single test type.

Data View Description

This data view would be very similar to the detail page that opens when a data point is selected in perf-o-matic. A fixed time range would be displayed. A tooltip is presented for each data point when the user mouses over it. The following is a list of differences:

1. A horizontal line indicating the position of the cutoff for the particular Talos test would be displayed on the chart.
2. Comments can be associated with specific data points.
3. Data points can be color coded by users to indicate when a changeset is backed out.
4. Changesets can be selected and collected to import into additional data views that except time ranges or changesets as input. A box-plot presentation would be available.

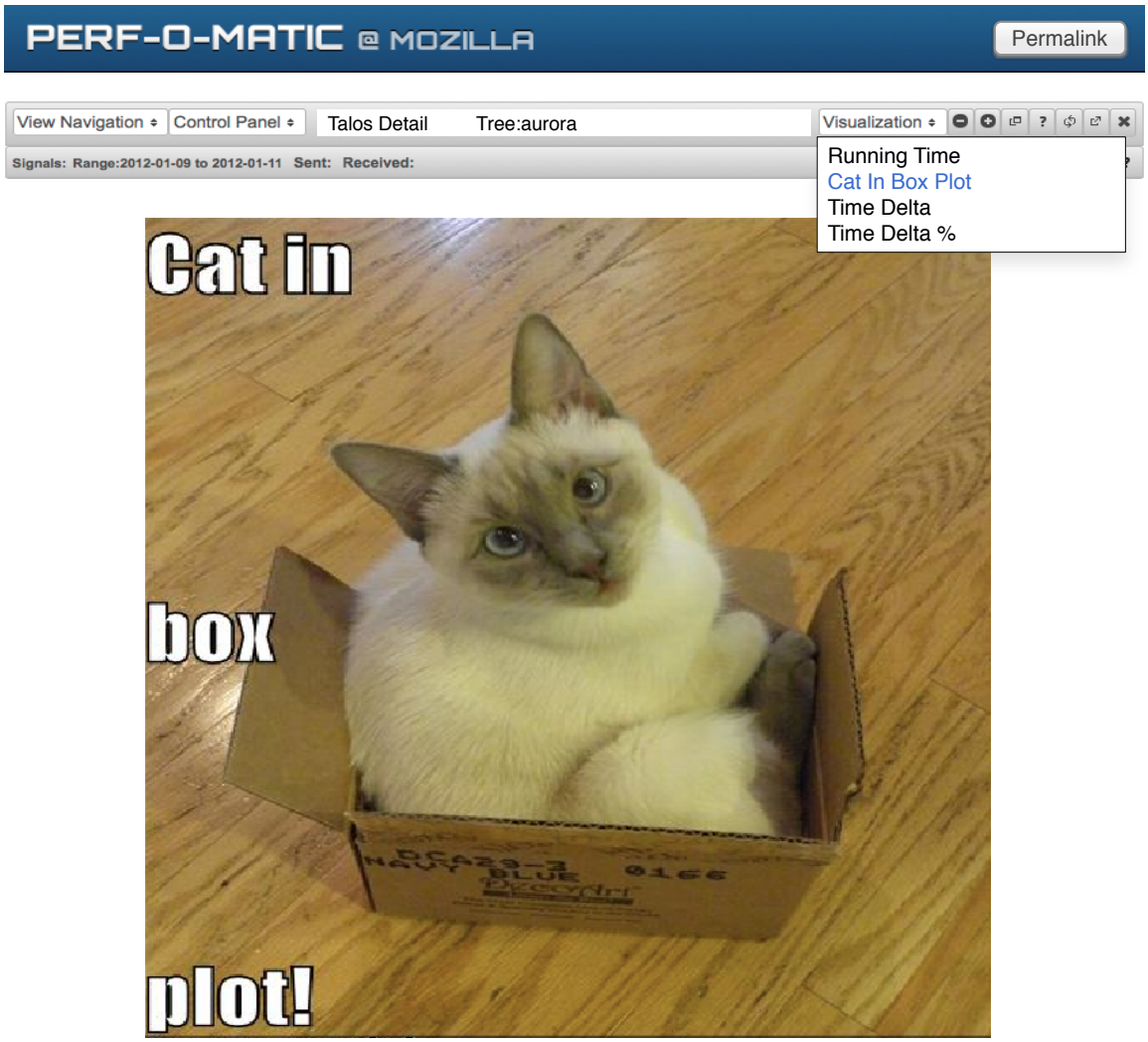
Talos Detail: Box Plot



Another way of examining the Talos Detail data would be using a box plot. Using the visualization menu, a user could select Box Plot; this would cause the content pane to display the new visualization. A box plot displays the minimum, maximum, median, mean, 20% quartile, and 75-80% quartile for every data point. We currently don't collect this information explicitly but might want to consider it. It would help understand what changes in the underlying data are causing shifts in mean values without having to display all of the raw values. If these values were explicitly stored there would be a variety of additional visualizations we could perform specifically looking for trends across changesets or a time range.

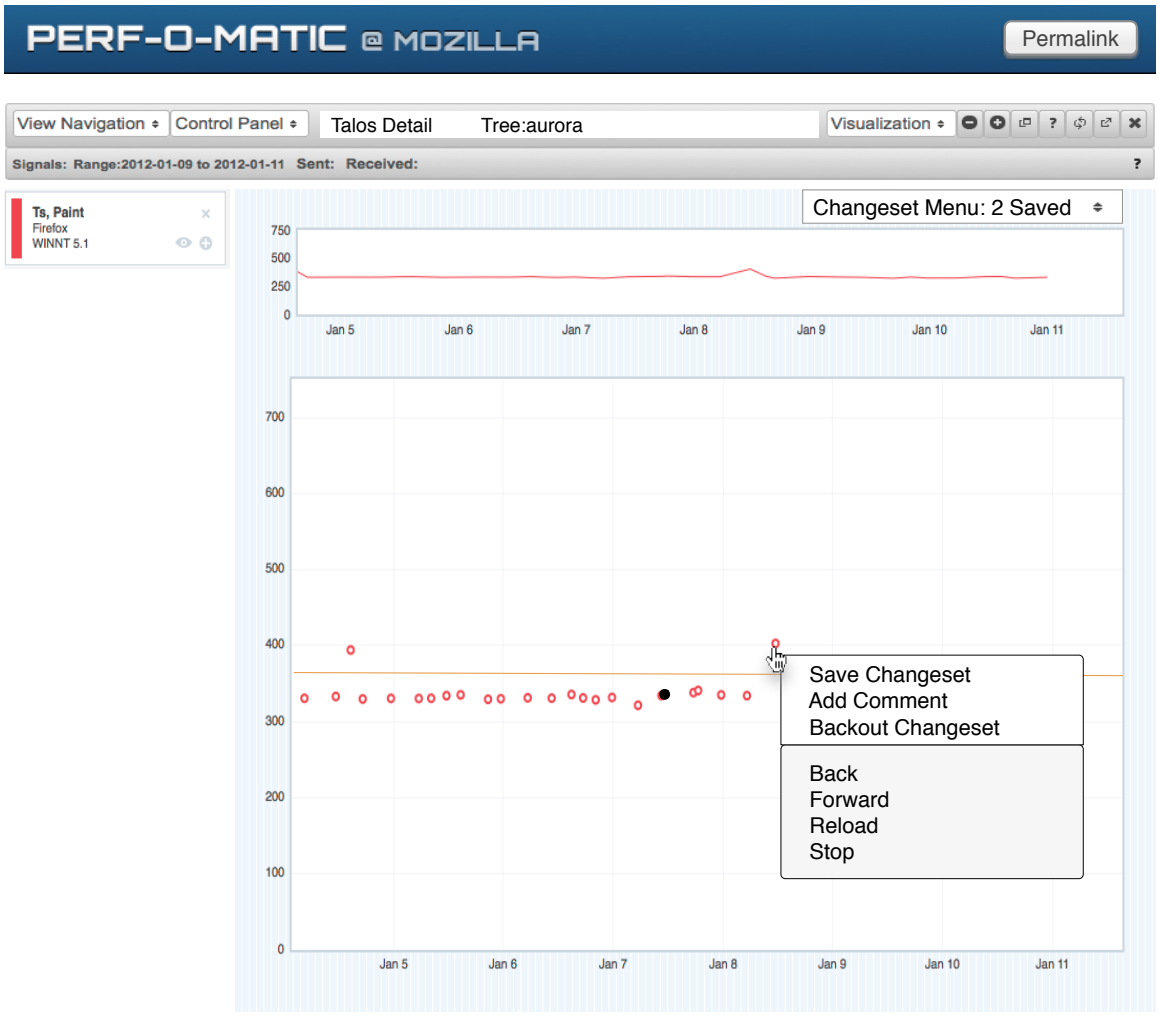
Talos Detail: Cat In Box Plot

Where there's a box plot there could be a Cat In Box Plot! The Cat In Box plot is a highly sought after and coveted data visualization tool. It's seldom used due to its immense complexity. If you find yourself thinking, wow this is a ridiculously long document, think how I must feel writing it. It's coming to an end soon, I promise, at least it has lots of pictures in it! Imagine if this was all text, it could be worse…

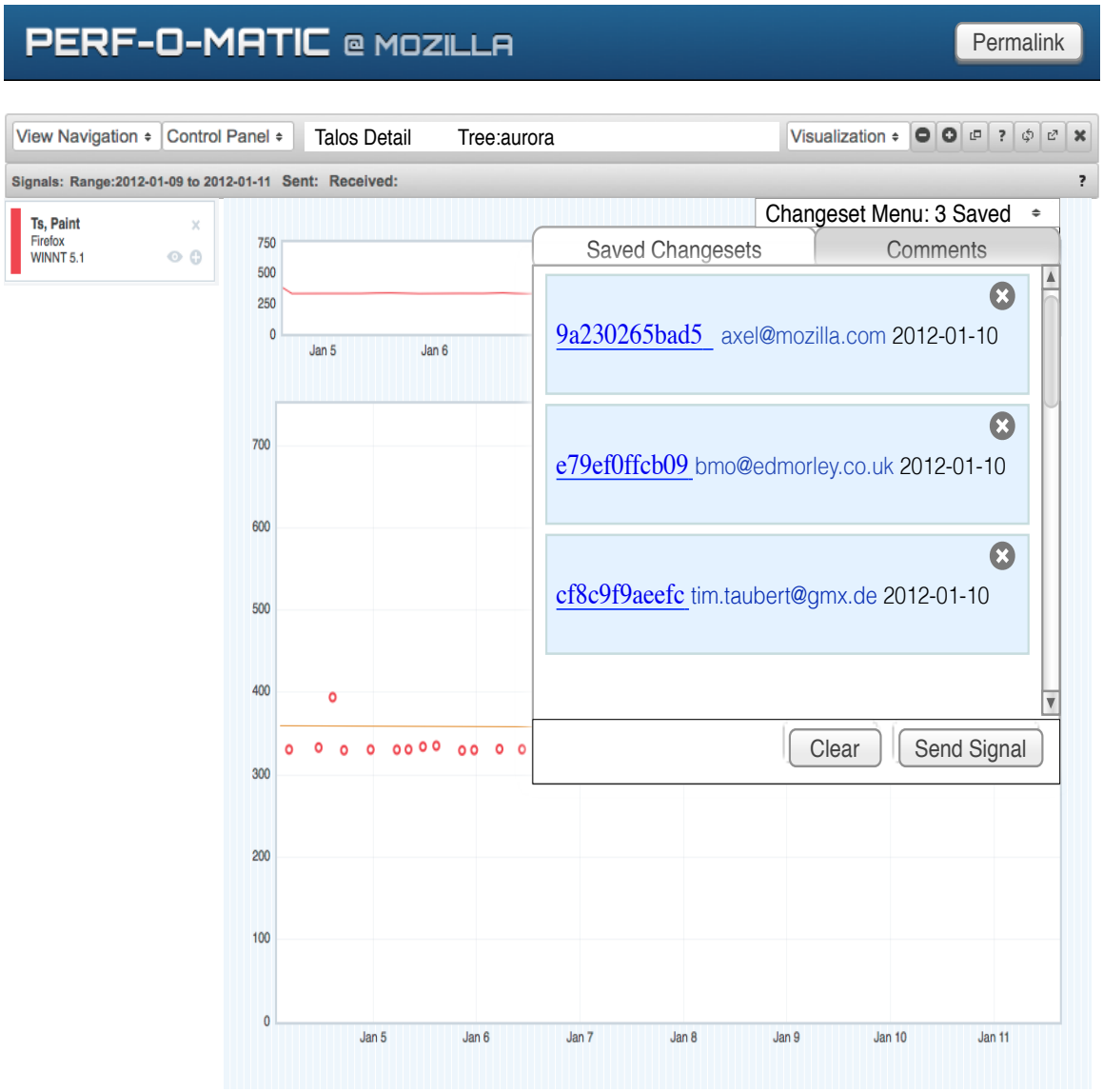


A right click context menu would be available for each data point. This context menu would present options for saving a changeset, adding a comment to a changeset, or indicating that a changeset was backed out.

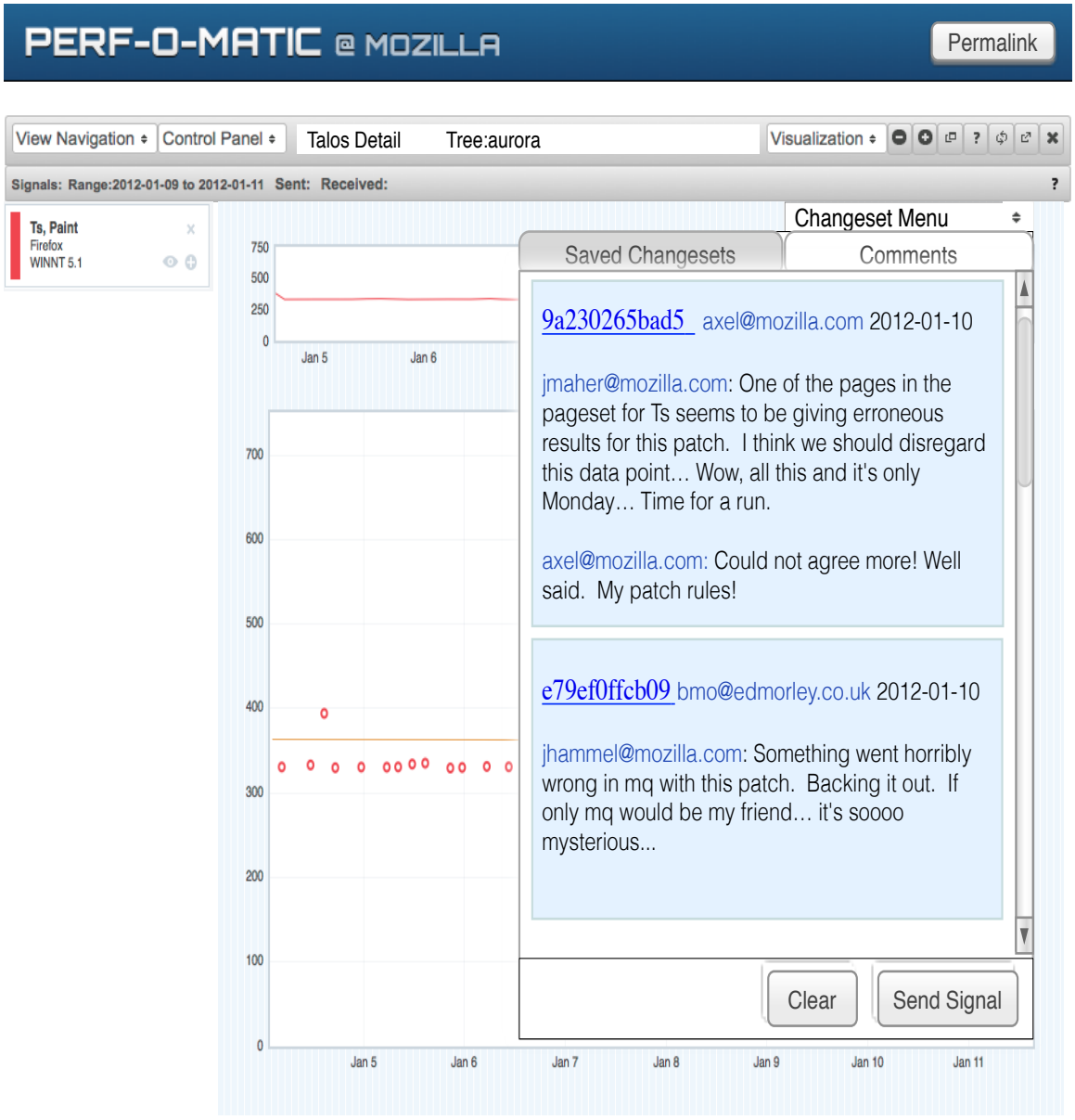
The Add Changeset option would open a modal that provides options for constructing a comment. This commenting system would help users persist their observations or add observations that no one else might be able to make. When a changeset has been backed out it is displayed as black and a user will be forced to add a comment.



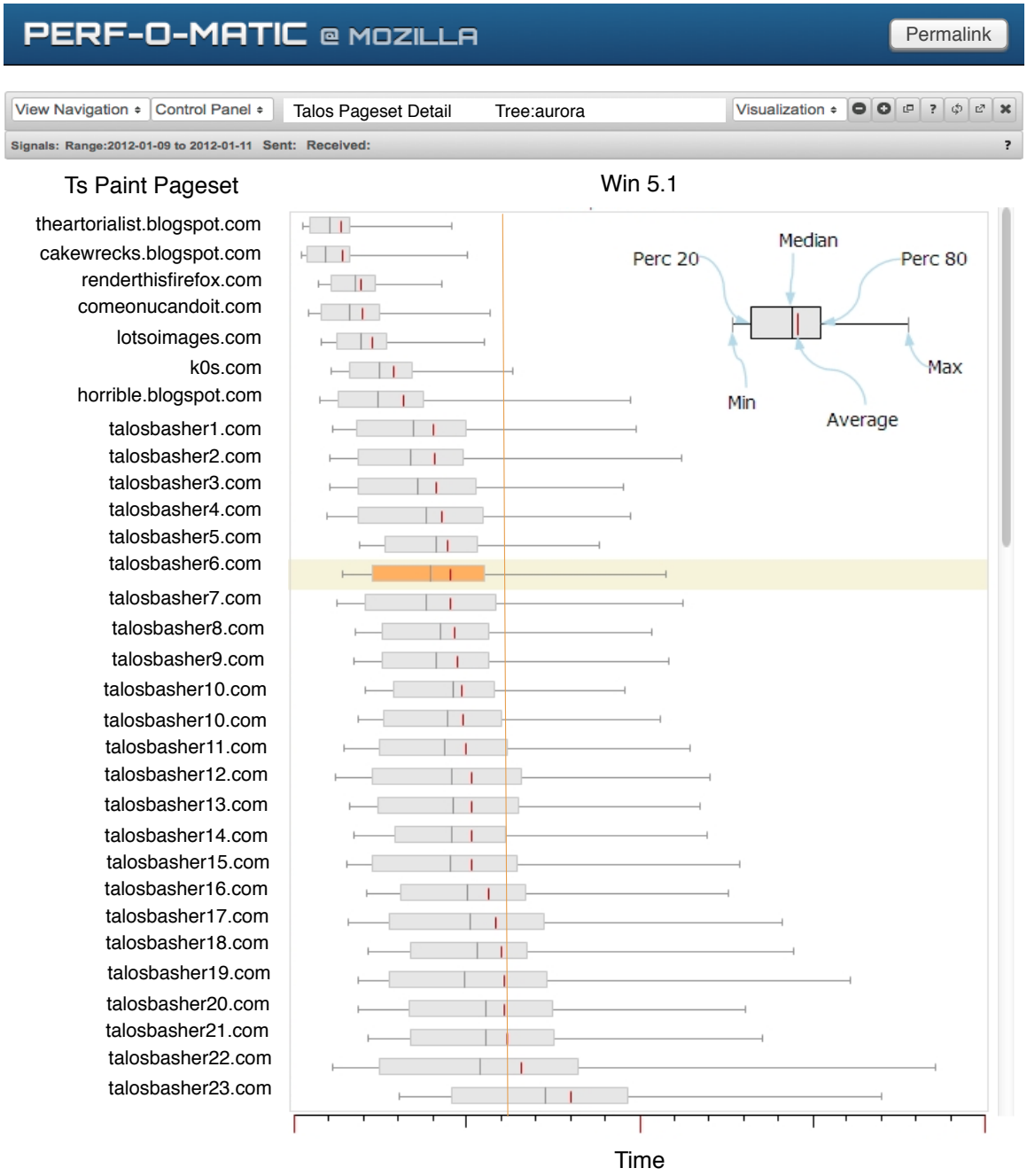
A dropdown Changeset Menu would be available. The dropdown menu would have two tabs. The Saved Changesets tab would show a list of changesets the user has selected using the context menu. Each changeset would have a delete option, which would remove the changeset from the saved list. The Send Signal button would send the changeset to any other data view that accepts a changeset or time range as a signal. The Clear button would remove the changests from the Saved Changesets list.



Comments would be visible in the Comments tab. This tab would be automatically opened when a user selects the comments link from the tooltip. A user could also browse there directly from the Changeset Menu and select the Comments tab. The comment selected would automatically scroll to the top of the dropdown menu. A changeset can have multiple comments from multiple authors.



Pageset Detail



Users: Firefox Developer, Talos Developer, Manager and Investigator

Entry Points: Application entry occurs by selecting a point or clickable element in the Talos Detail page that sends a signal containing a date range, source tree, Talos test, platform, and changest id.

Use Cases:

1. User makes an observation using Talos Detail and wants to explore the underlying data.
2. User wants to see the data distribution of a Talos test pageset.
3. User observes an outlier and wants to determine which pageset pages were responsible for the fluctuations in the mean value.
4. User modifies a Talos test and wants to see the pageset results.

Data View Description

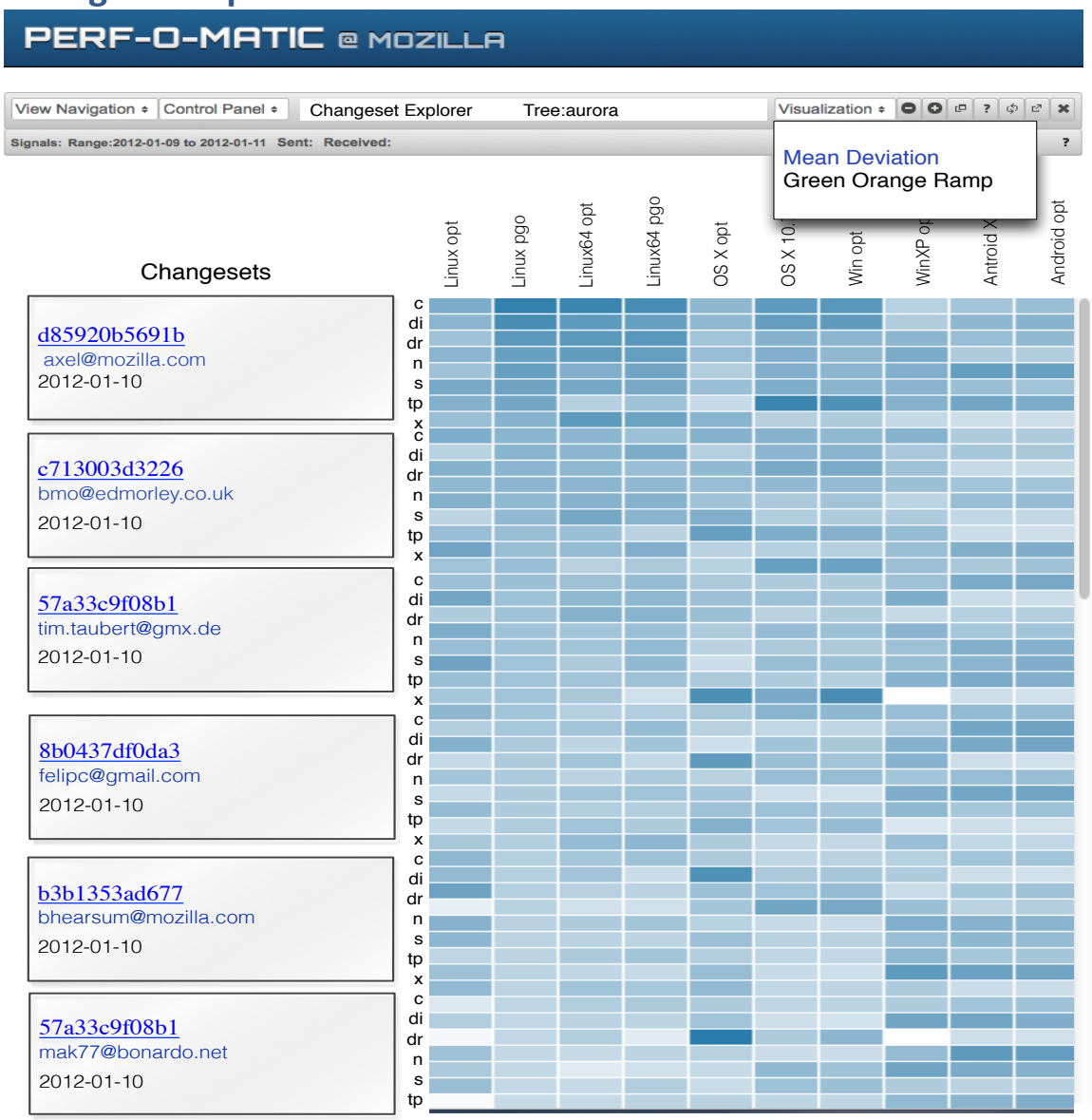
The y-axis contains each web page in a pageset for a particular test. The x-axis displays time associated with the pageset run using box plots showing the minimum, maximum, median, mean, 20% quartile, and 75-80% quartile for all repetitions for a particular changeset. The orange vertical line would be the cutoff for this particular test and pageset combination.

One could compare pages in a pageset across multiple platforms by adding platform vertical panels.

This same representation could be shown as a heat map or space-filling tree. One could also compare pagesets across different changesets by adding more y-

axis page entries and some labels. I will spare everyone including myself the mocks for that.

Changeset Explorer



Users: Firefox Developer, Talos Developer, Sheriff, Manager, and Investigator

Entry Points: Application entry occurs by saving changesets in the Talos Detail page and then clicking send signal on the Changesets dropdown menu. The signal data would include the date range, source tree, Talos test, platform, and list of changeset ids.

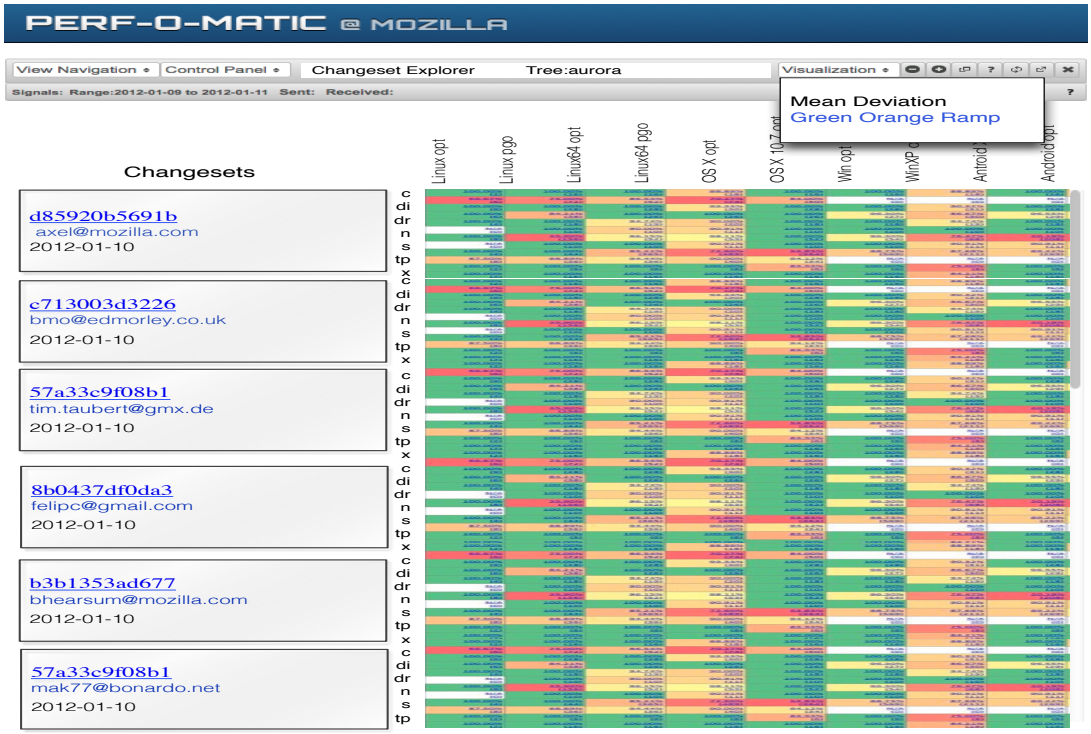
Use Cases:

1. User wants to explore a group of changeset outliers, looking for any differences/similarities in their test results.
2. User wants to see if a trend is occurring across a set of changesets.
3. User observes an outlier and wants to determine if other Talos tests are fluctuating.

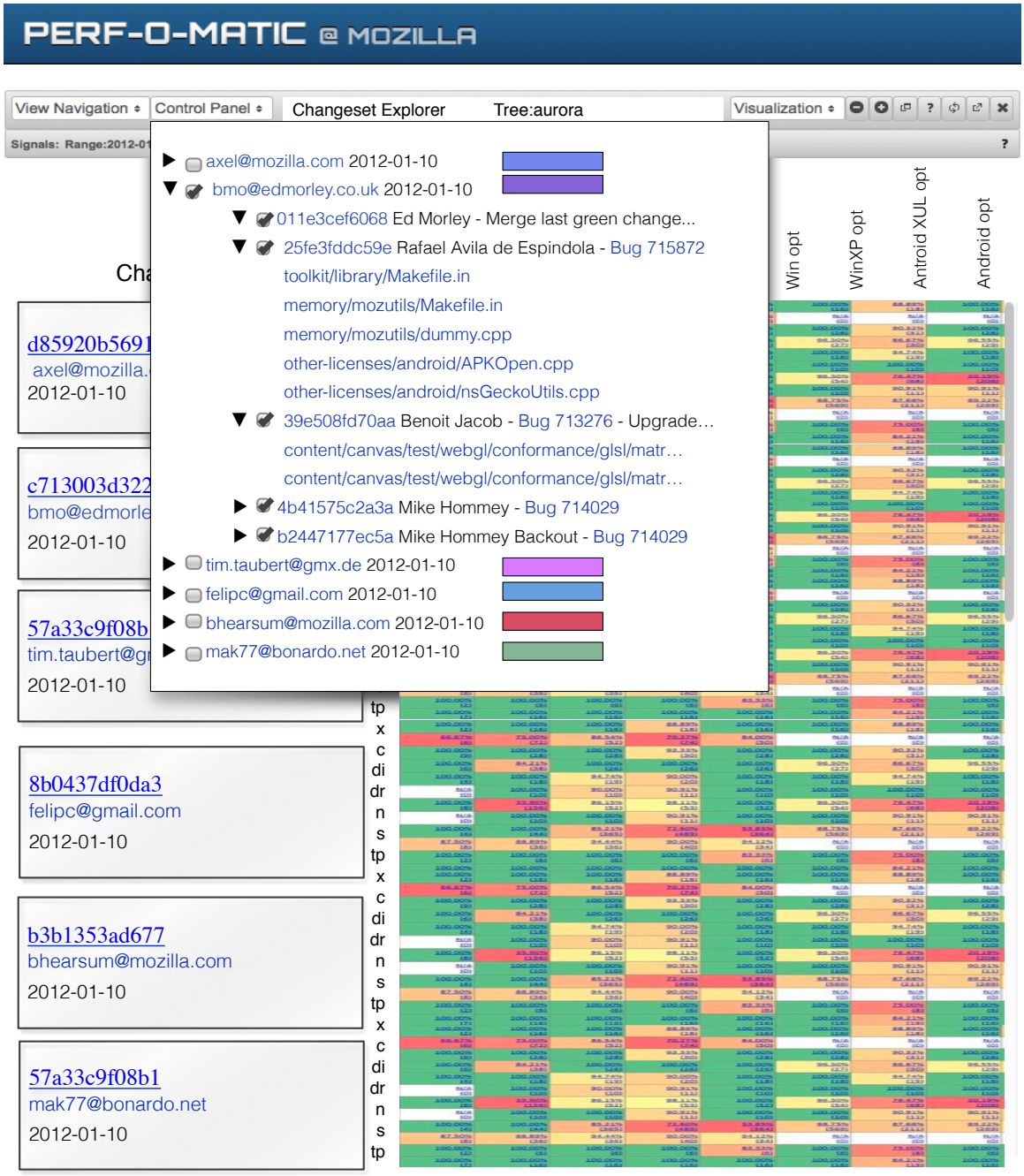
Data View Description

Another possible visualization that would help assess trends across different changesets is a heat map. In the heat map, Talos tests are laid out on each row while each column specifies a different platform. The gray boxes on the left indicate the changeset associated with the test results, allowing for the comparison of multiple changesets or just a single changeset. Each cell would represent a mean of means of median values that could be compared to the cutoff indicating test failure to determine what color should be assigned.

The example shown above is colored by Mean Deviation which would be a qualitative indicator of which changest, test, and platform combinations show the most variation. The same green to orange color ramp described in Talos Regression Hunter could also be applied to this visualization. This is shown below. The color ramp would provide a convenient way to assess how tests relate to the cutoff.



The control panel could contain a tree of all of the changesets. This tree would allow a user to add/remove changesets from the visualization. It could also enable a user to browse the changeset file changes.

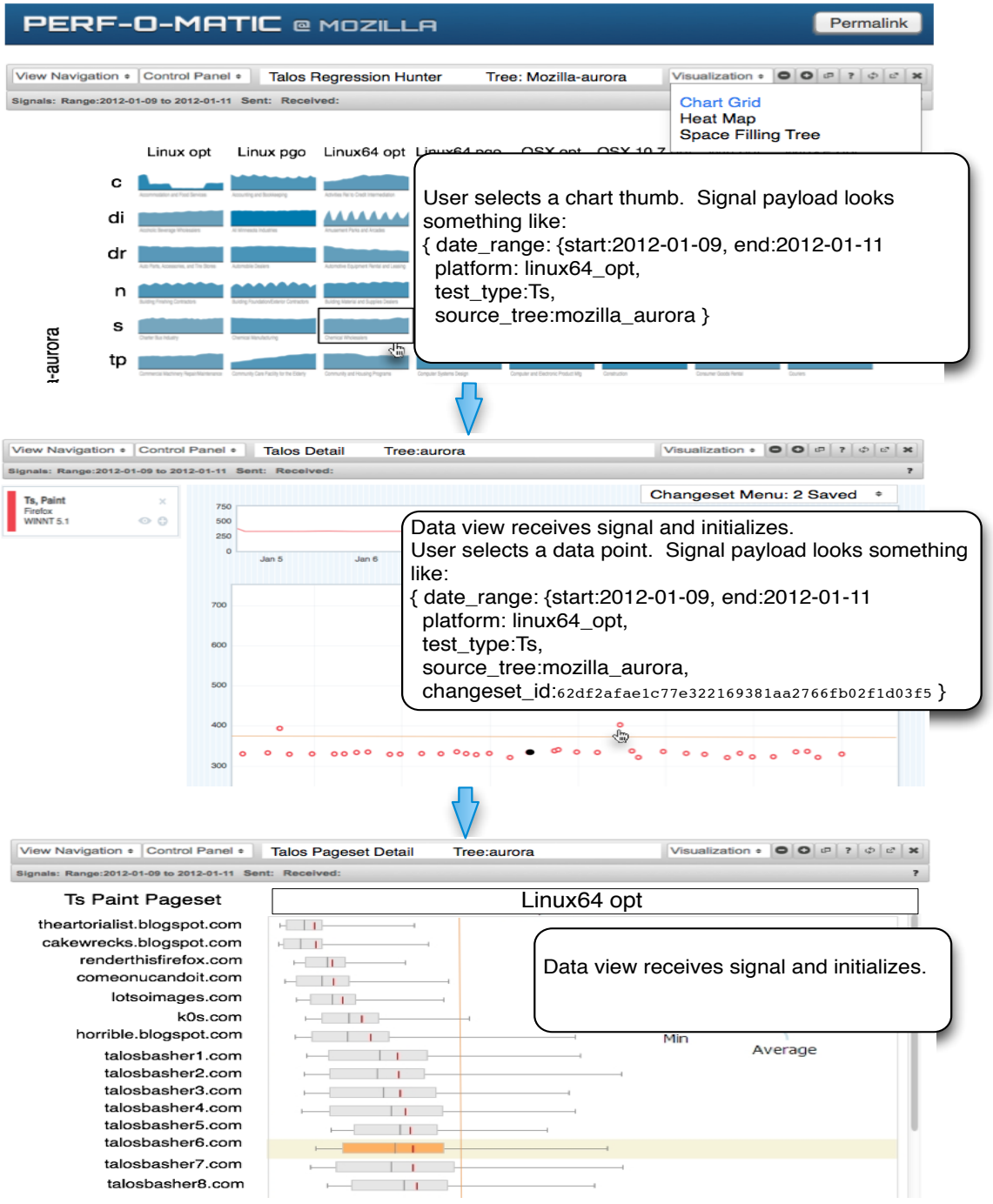


One additional use case for this might be maintaining context when entering the application from TBPL. The changeset explorer could display the changeset tree as it appears in TBPL for a given source tree. This might help a Sheriff or Investigator maintain context between the two applications when examining a potential trend in results.

Data View Collections

Talos Collection

Putting all of these data views together with the signaling concept allows a user to rapidly explore varied forms of connected data. The following mock shows three of the data views described: Talos Regression Hunter, Talos Detail, and Talos Pageset Detail working together as a collection within the same page.

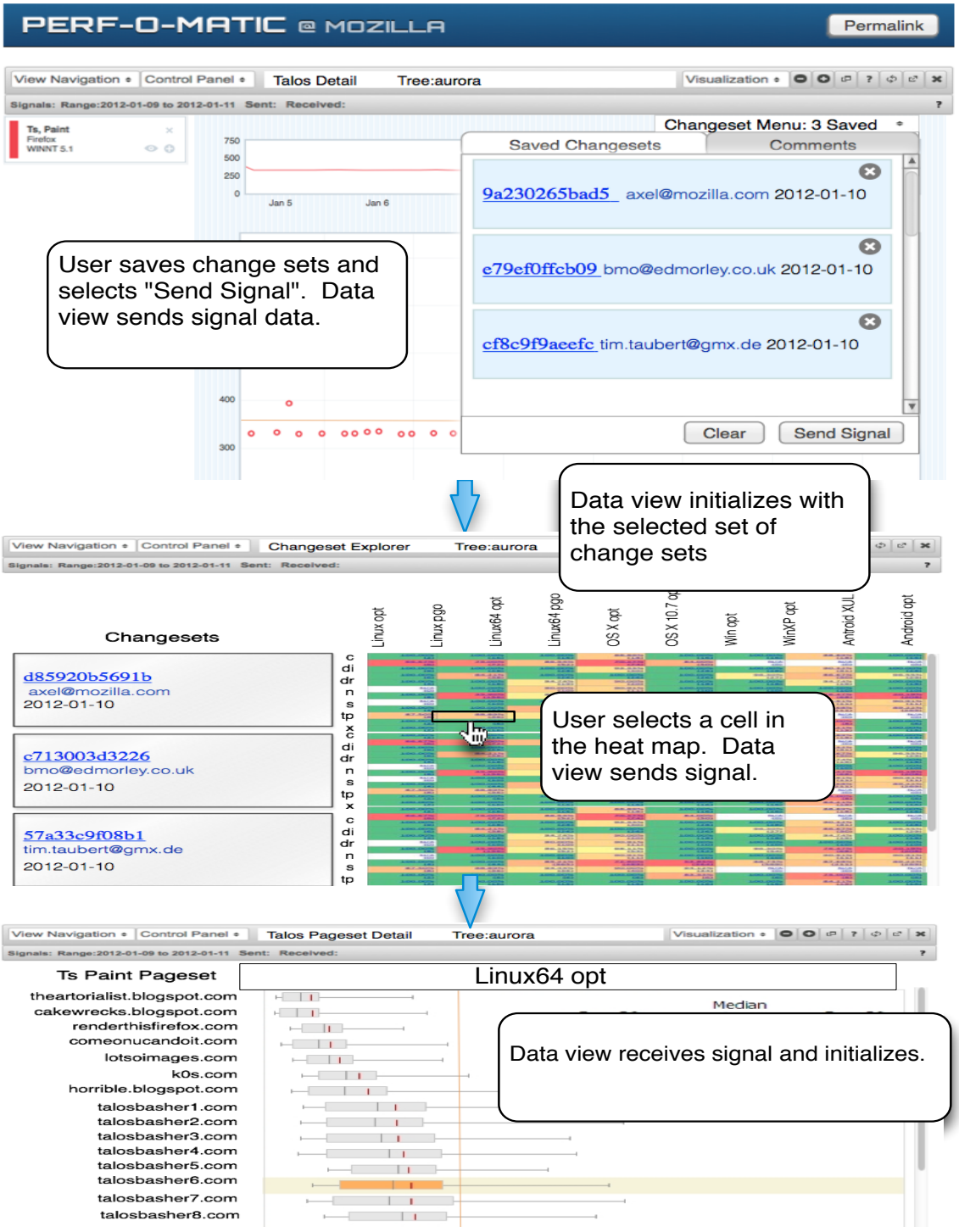


Using this signaling strategy a user could very rapidly browse the data associated with many of chart thumbs in Talos Regression Hunter while keeping context with pageset data to help answer detailed questions.

This is the same collection shown before but the user has selected alternative visualizations. The alternate visualizations send the same types of signals.

Talos Detail Collection

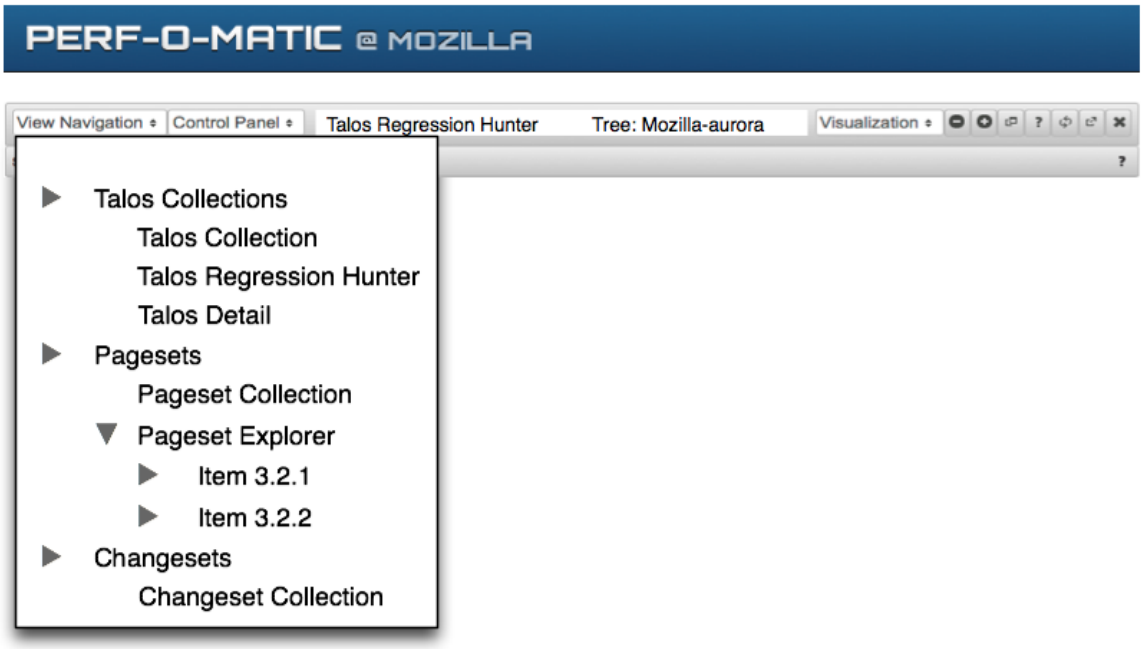
This collection includes Talos Detail, Changeset Explorer, and Talos Pageset Detail.



A user could rapidly scan and collect changesets in Talos Detail that are outliers or are interesting for some reason and then select Send Signal to initialize Changeset Explorer to the changesets of interest. This would allow the user to see the changeset results across all platforms. A user could drill down further by

selecting a heat map cell and initializing Talos Pageset Detail to the actual pageset associated with a test. This could be done in rapid succession for many changesets.

Data View Collection Navigation

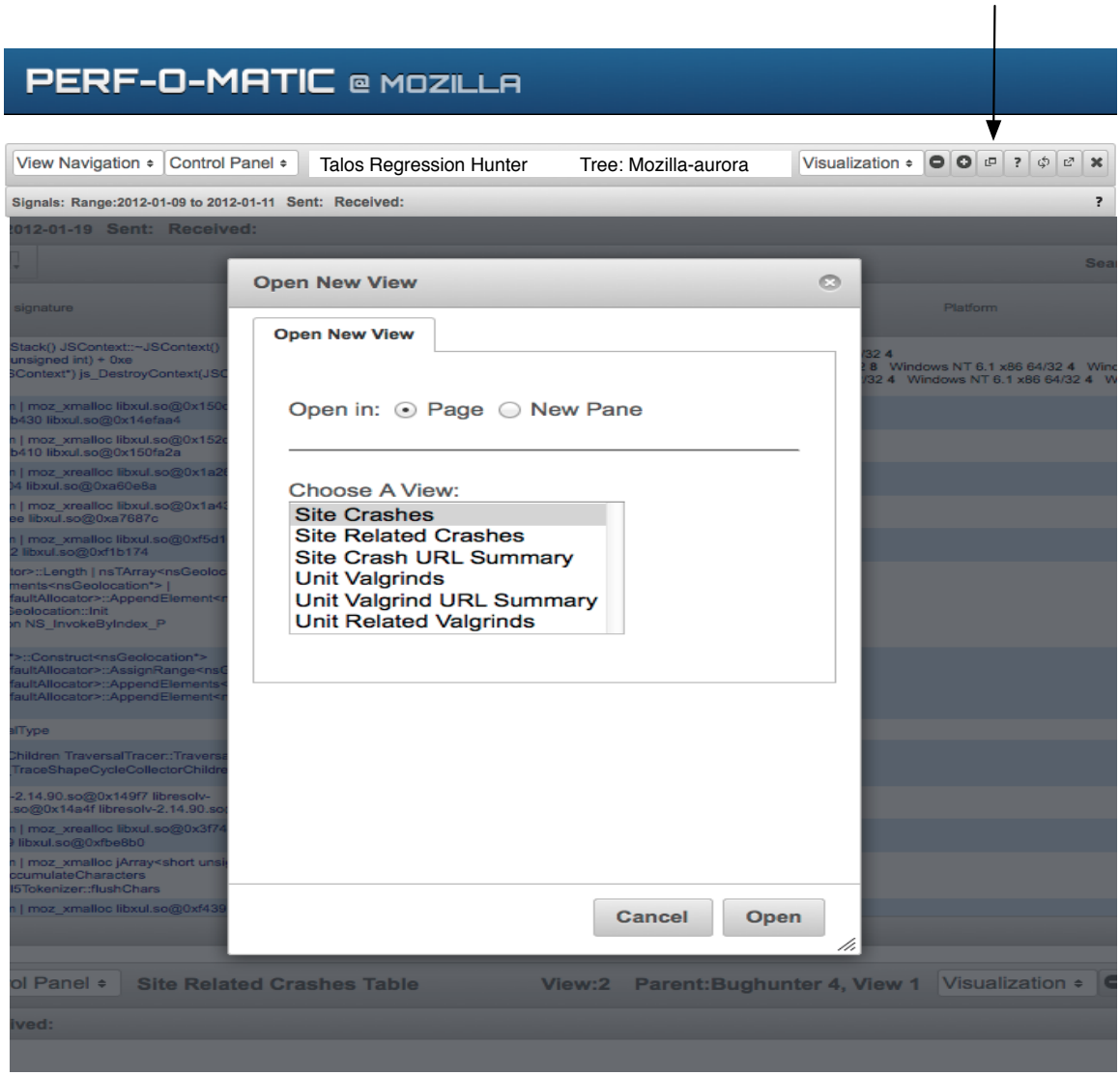


The View navigation menu would allow a user to change a data view to another view or load a collection of data views that work together. Collections would allow the application to address the 80% use case, these collections would be sets of data views that signal each other and address some of the more common use cases. Data view collections could be created around mobile tests or desktop tests, maybe that would be a good entry point for some use cases.

This navigation would scale to manage many different forms of data. Maybe different tests in addition to Talos could eventually be incorporated.

Custom Collections

User selects the Open New View button



Users could build or extend existing collections by selecting the Open New View button and choosing a data view to incorporate into the current page or a new page if they have the monitor space. This new view would be a child of the data view that opened it and would automatically receive signals from it.

W00t! You made it, kudos to you, now go reward yourself with your favorite vice!