Analytics in the Sun 7000 Series

Bryan Cantrill, Brendan Gregg

Sun Microsystems Fishworks

The Problem

Storage is unobservable

- Historically, storage administrators have had very little insight into the nature of performance, with essential questions largely unanswerable:
  - "What am I serving and to whom?"
  - "And how long is that taking?"
- Problem is made acute by the central role of storage in information infrastructure – it has become very easy for applications to "blame storage"!
- It has therefore become up to the storage administrator to exonerate their infrastructure – but limited toolset makes this excruciating/impossible

The Problem

But wait, it gets worse

- Those best positioned to shed some light on storage systems are those with the greatest expertise in those systems: the vendors
- But the vendors seem to have the same solution for every performance problem:
  - Buy faster disks ($$$)
  - Buy more, faster disks ($$$ · n)
  - Buy another system ($$$ · n + $$$$)
  - Buy another, bigger system ($$$ · n + $$$$$$$$)
- This costs the customer a boatload – and doesn't necessarily solve the problem!

Solving the Problem

Constraints on a solution

- Need a way of understanding storage systems not in terms of their *implementation*, but rather in terms of their *abstractions*
- Must be able to quickly differentiate between problems of *load* and problems of *architecture*
- Must allow one to quickly progress through the *diagnostic cycle*: from hypothesis to data, and then to new hypothesis and new data
- Must be *graphical* in nature – should harness the power of the visual cortex
- Must be *real-time* – need to be able to react quickly to changing conditions

Envisioning a Solution

Implementation versus abstraction

- **The system's implementation – network, CPU, DRAM, disks – is only useful when correlated to the system's abstractions**
- **For a storage appliance, the abstractions are at the *storage protocol* level, e.g.:**
  - ***NFS operations* from *clients* on *files***
  - ***CIFS operations* from *clients* on *files***
  - ***iSCSI operations* from *clients* on *volumes***
- **Must be able to instrument the protocol level in a way that is semantically meaningful!**

Envisioning a Solution

Architecture versus load

- Performance is the result of a given *load* (the work to be done) on a given *architecture* (the means to perform that work)
- One should not assume that poor performance is the result of inadequate architecture; it may be due to inappropriately high load!
- The system cannot automatically know if the load or the architecture is ultimately at fault
  - The system must convey *both* elements of performance
  - The decision as to whether the problem is due to load or due to architecture must be left as a *business* decision: administrator must either *do less* or *buy more*

Envisioning a Solution

Enabling the diagnostic cycle

- **The diagnostic cycle is the progression from hypothesis through instrumentation and data gathering to a new hypothesis:**

  hypothesis → instrumentation → data → hypothesis

- **Enabling the diagnostic cycle has implications for any solution to the storage observability problem:**
  - **System must be *highly interactive* to allow new data to be quickly transformed into a new hypothesis**
  - **System must allow *ad hoc* instrumentation to allow instrumentation to be specific to the data that motivates it**

Envisioning a Solution

Engaging the visual cortex

- The human brain has evolved an extraordinary ability to visually recognize patterns
- Tables of data are not sufficient – we must be able to *visually represent data* to allow subtle patterns to be found
- This does *not* mean merely "adding a GUI" or bolting on a third-party graphing package, but rather *rethinking* how we visualize performance
- Visualization must be treated as a *first-class* aspect of the storage observability problem

Envisioning a Solution

Need real-time interaction

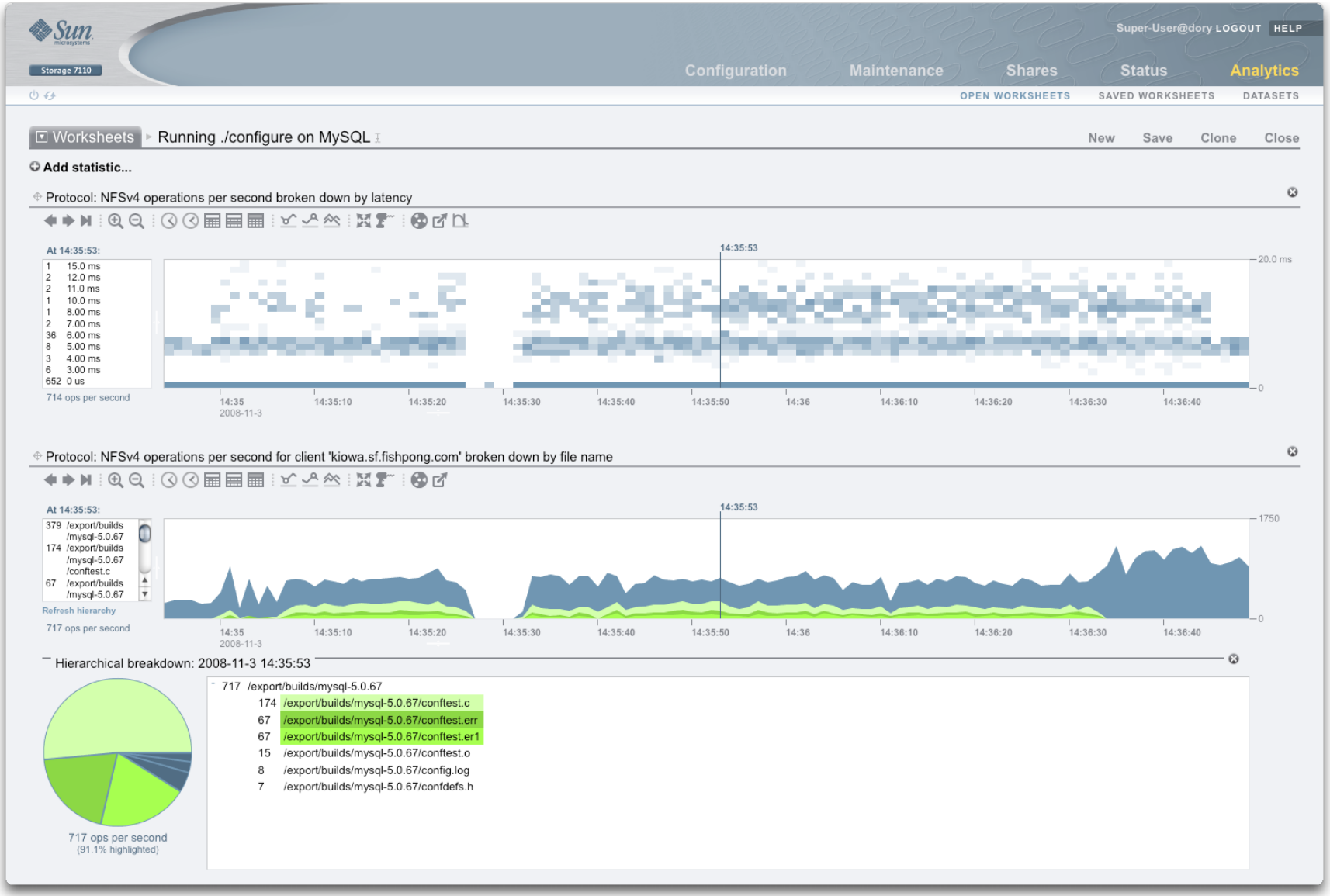
- Post-facto analysis tools suffice for purposes such as capacity planning, when time scales are on the order of purchasing cycles and the system is not pathological...
- ...but such tools are of little utility when phones are ringing and production applications are degrading
- The storage administrator needs to be able to interact with the system in *real-time* to understand the dynamics of the system
- Need to be able to understand the system at a fine temporal granularity (e.g., one second); coarser granularity only clouds data and delays response

Towards a Solution

DTrace: a tantalizing foundation

- DTrace is a multiplatform (& award-winning!) facility for the dynamic instrumentation of production systems
- DTrace excels at cutting through implementation to get to the semantics of the system
- DTrace has proven ability to separate architectural limitations from load-based pathologies
- DTrace is but foundation:
  - Still need abstraction layer above programmatic interface
  - Still need mechanism to visualize data
  - Still need the ability to (efficiently!) store historical data

Introducing Appliance Analytics



Appliance Analytics

"Your AJAX fell into my DTrace!"

- DTrace-based facility that allows administrators to ask questions phrased *in terms of storage abstractions*:
  - "What clients are making NFS requests?"
  - "What CIFS files are being accessed?"
  - "What LUNs are currently being written to?"
  - "How long are CIFS operations taking?"
- Data is represented *visually*, with the browser as vector
- All data is *per-second* and available in *real-time*
- Data is optionally recorded, and can be examined historically

Appliance Analytics

Ad hoc queries

- The power of analytics is the ability to formulate *ad hoc* real-time queries based on past data:
  - "What files are being accessed by the client 'kiowa'?"
  - "What is the read/write mix for the file 'usertab.dbf' when accessed from client 'deimos'?"
  - "For writes to the file 'usertab.dbf' from the client 'deimos' taking longer than 1.5 milliseconds, what is the file offset?"
- The data from these queries can *themselves* be optionally recorded, and the resulting data can become the foundations for more detailed queries

Analytics Overview

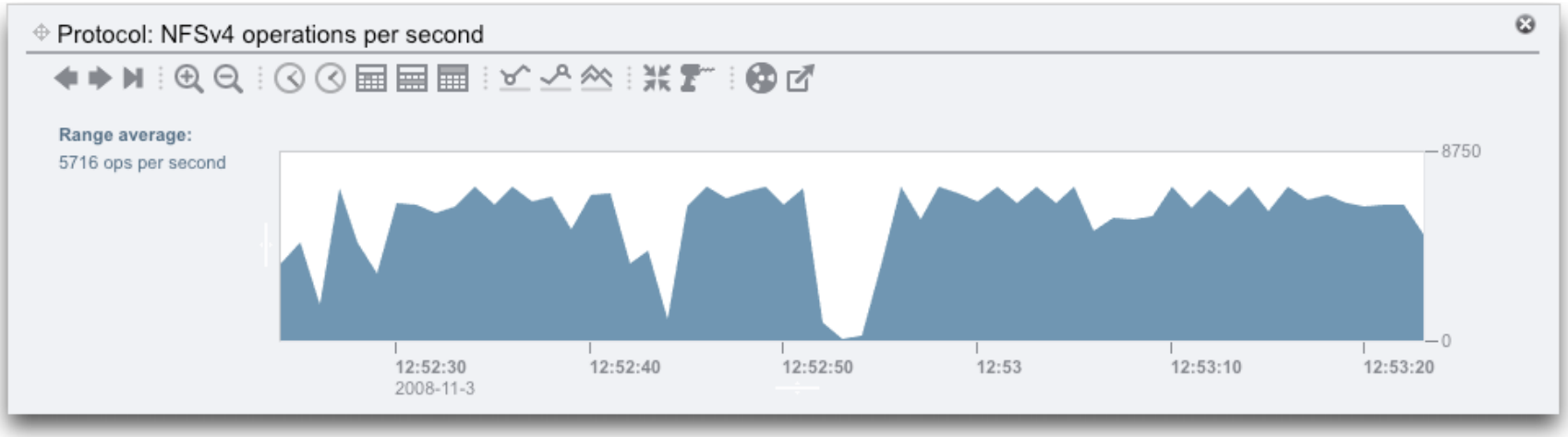
Statistics

- Analytics display and manipulate *statistics*
- A statistic can be a *raw statistic* – a scalar recorded over time (e.g., “NFSv3 operations per second”)
- Statistics can also be *broken down* into their constituent elements (e.g., “NFSv3 operations per second broken down by client”)
- To add a statistic, click on the “Add Statistic...” button
- A pop-up menu will appear:
  - Select statistic of interest by clicking on it
  - A cascading menu will appear with break down options
  - Select dimension in which to break down (if any)

Analytics Overview

Graphing statistics

- **Once a statistic has been selected, a new panel is added to the display, containing a graph of the statistic, updated in real-time:**

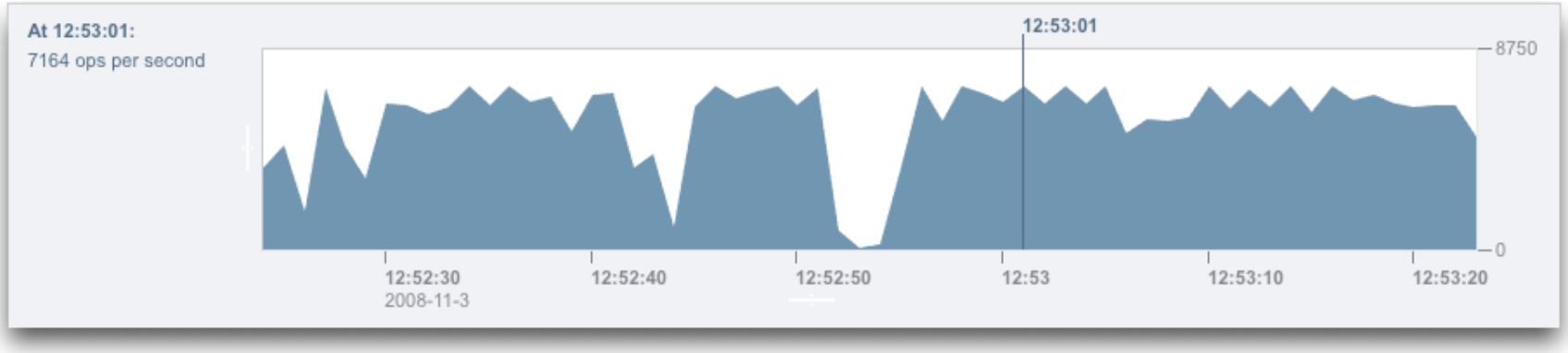


- **Time (in browser's locale) is on X axis; value is on Y axis**
- ***Average* over interval is displayed to left of graph**

Analytics Overview

Value at a moment in time

- To get the value of a statistic at a particular time, click on that time *in the graph*
- A bar will appear, labelled with the time, and the display to the left of the graph will change to be the value at the time selected:

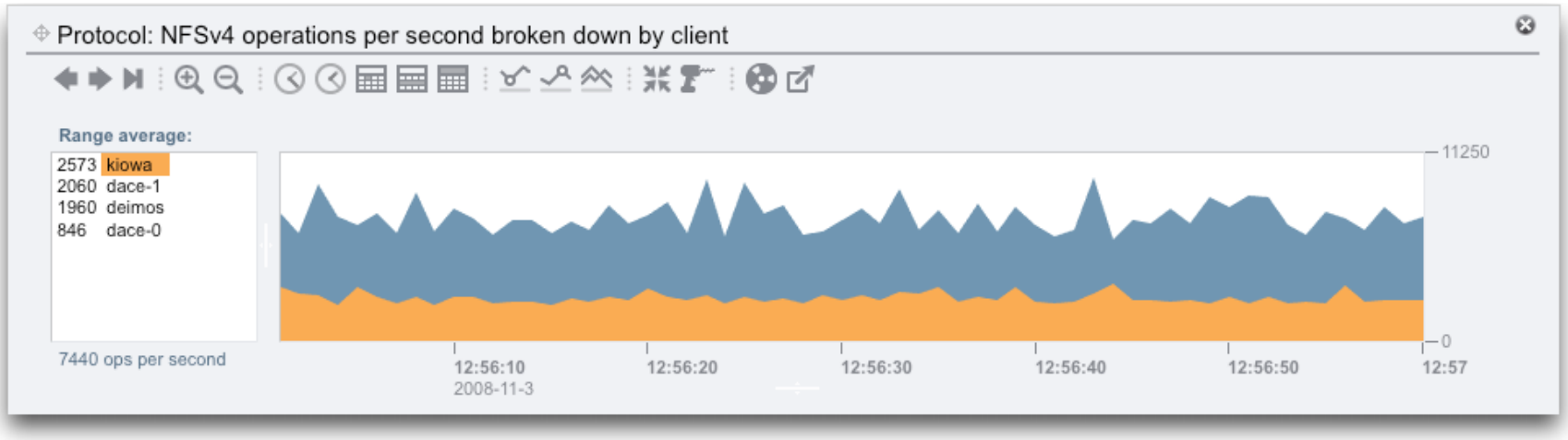


- Bar will move as graph updates in real-time – and note that the time will stay selected if it moves out of view!

Analytics Overview

Breaking down statistics

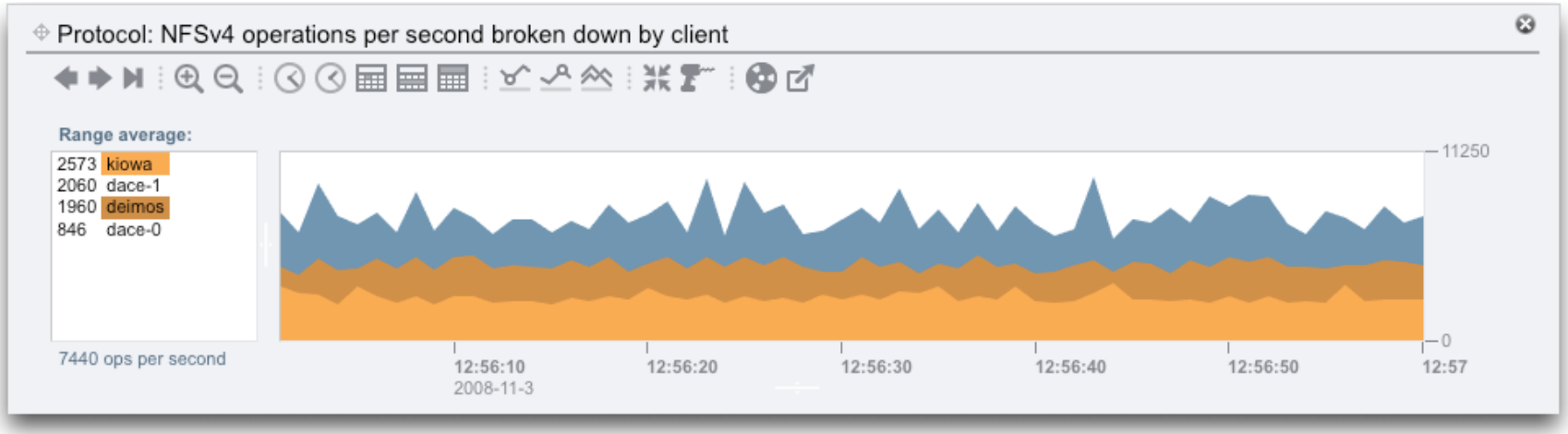
- For breakdown statistics, the area to the left of the graph contains a *breakdown table* showing average value of each element
- To see one element of a breakdown in the graph, click on its entry in the table:



Analytics Overview

Breaking down statistics

- To see multiple elements of a breakdown, click on one element and then shift+click on the others:

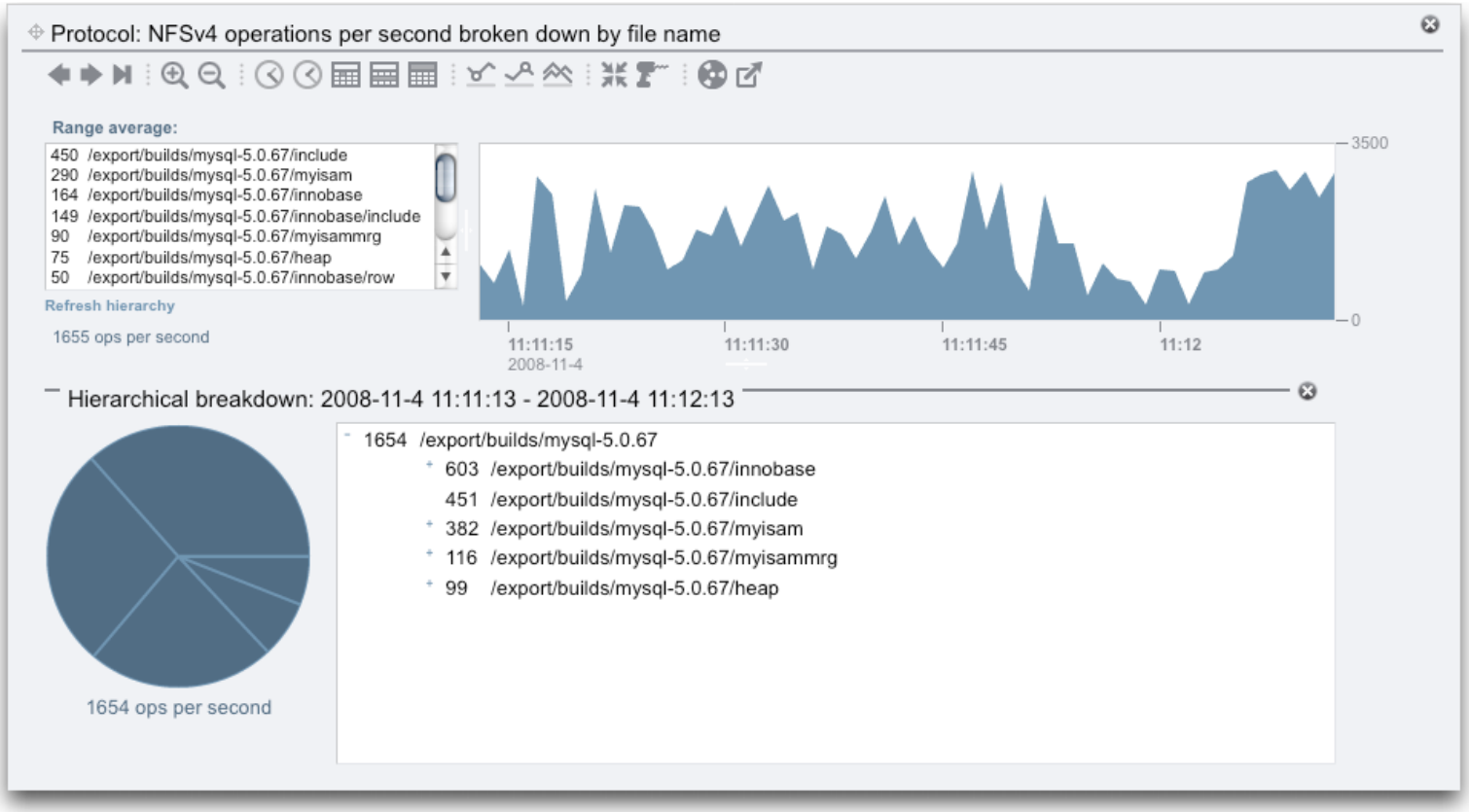


- The table consists of the top ten elements over the displayed time period; if more elements are available ellipsis (“...”) will appear as last element in table
- Click on ellipsis to see additional elements

Analytics Overview

Hierarchical breakdowns

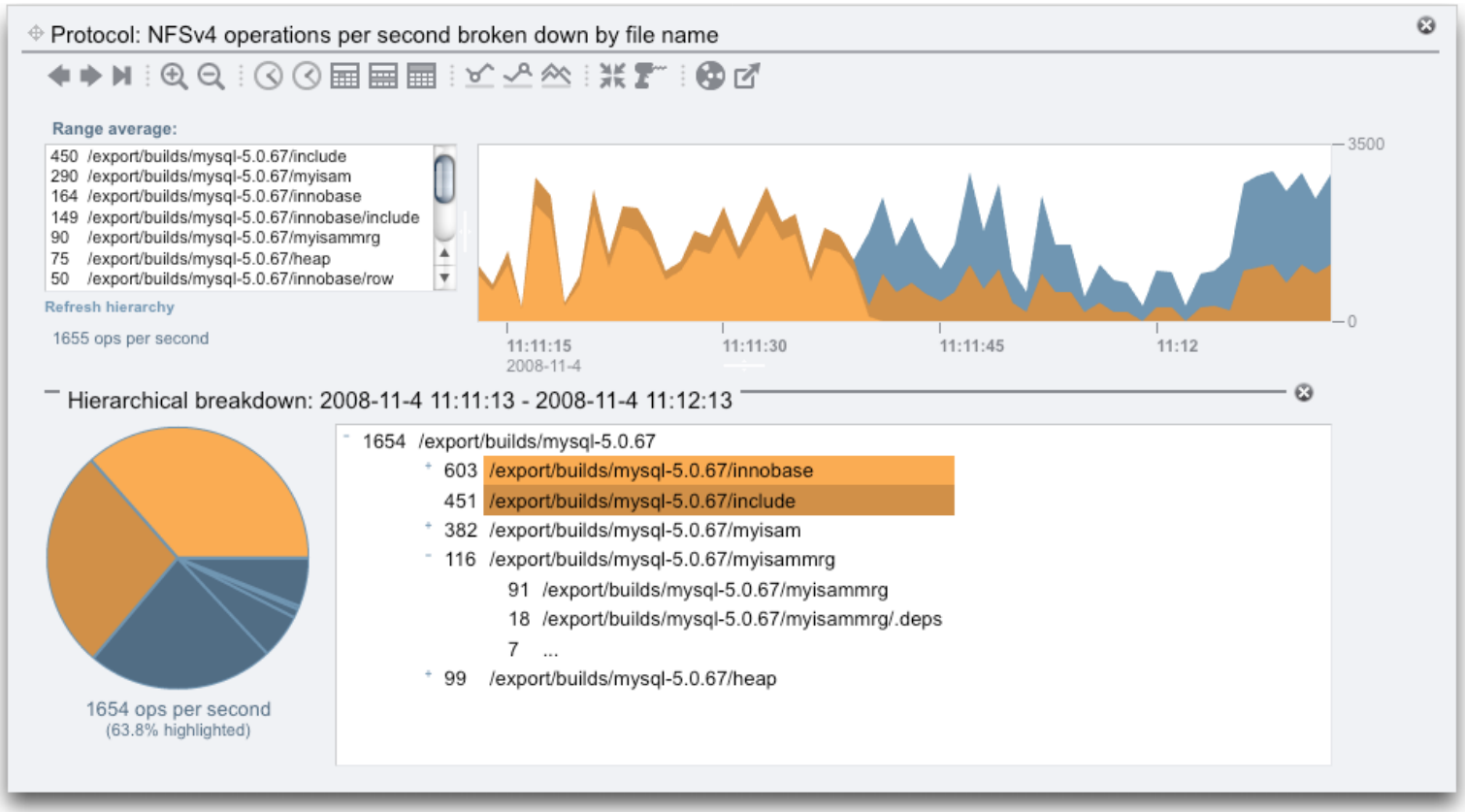
- **For files and devices, can visualize hierarchically by clicking “Show hierarchy” under breakdown table:**



Analytics Overview

Hierarchical break downs

- **Expand hierarchy by clicking on plus (“+”) button; highlight breakdown in graph/chart by clicking on text:**



Analytics Overview

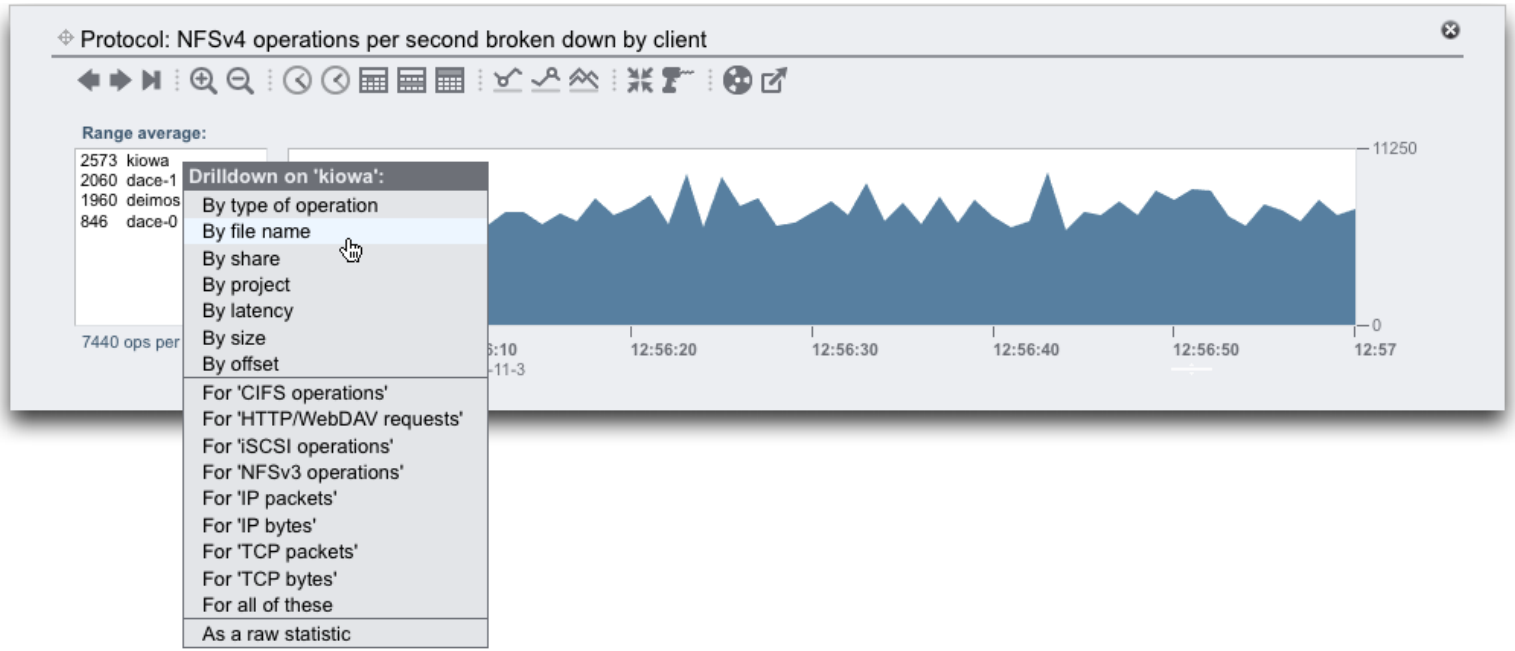
Hierarchical breakdowns

- Can also highlight a breakdown by *clicking on a wedge* in the pie chart
- Hierarchical breakdowns are *not* automatically updated when the graph is updated!
  - When a breakdown is extensive, calculating the hierarchical breakdown can be expensive
  - The label on the hierarchical breakdown has the time/date range for which the breakdown applies
  - To refresh the hierarchical view, click “Refresh hierarchy” below the breakdown table

Analytics Overview

Drilling down on statistics

- ***Ad hoc*** **queries are formed by** ***drilling down*** **on a particular element in a broken down statistic**
- **To drill down on a particular element,** ***right click*** **on it, and then select a new breakdown:**



Analytics Overview

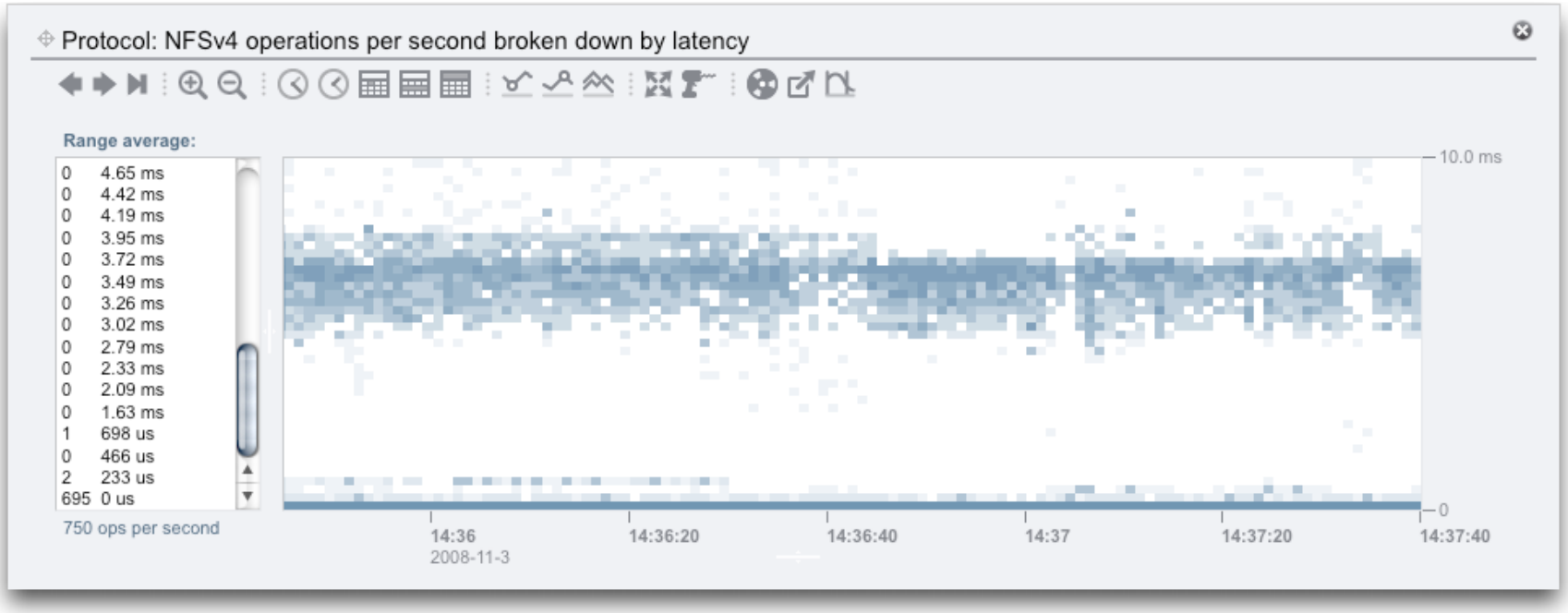
Quantized breakdowns

- For some statistics – e.g. operation latency, size, offset, etc. – a scalar is not sufficiently expressive:
  - Average can be highly misleading
  - Zero-valued data must be distinguished from no data
- For these operations, need to understand the *distribution* of data over time – need a *histogram* per unit time
- Analytics allows this via *quantized breakdowns*

Analytics Overview

Graphing quantized breakdowns

- **Graph for quantized breakdown consists of time on X axis, values on Y axis, and a *heat map* (a color-coded histogram) per sample:**



Analytics: Controlling the graph

- Each graph has a button bar that controls the graph's attributes
- Many buttons have auxiliary functionality available by *shift-clicking*: holding down Shift while clicking on button

Graph Controls: Moving and pausing

- By default, the latest represented time in a graph is the current time (i.e., graphs reflect data up to the present)
- Graphs are automatically updated in real-time such that this remains true
- Use the arrow buttons to navigate historically:
  - Left arrow moves represented time towards the past
  - Right arrow moves represented time towards the present
- The pause button suspends real-time updates to the graph – but does *not* suspend collection of the underlying data!

Graph Controls: Zooming in and out

- Zoom buttons change the *time scale* of the graph
- Zoom does *not* change the resolution of gathered data; data is *always* gathered at one second granularity
- Cannot zoom in any further than five seconds in a graph
- Can zoom out arbitrarily – to years, if so desired
- When zoomed out sufficiently far that multiple samples are represented in a single pixel, the *maximum* of those samples is treated as the representative to visualize
  - Low values get "drowned out" when zoomed out
  - Undesirable, but alternative seems worse: spikes would "appear" as one zooms in

Graph Controls: Time scale shortcuts

- Zooming is tedious for radical changes to time scale
- Time-based buttons offer convenient shortcut:
  - Minute button sets time scale to one minute
  - Hour button sets time scale to one hour
  - Day button sets time scale to one day
  - Week button sets time scale to one week
  - Month button sets time scale to one month
- Shift-click on time-based button *adds* that unit of time to the time scale
  - Clicking on minute button sets time scale to one minute; shift-click on minute button sets scale to two minutes

Graph Controls: Minimum and maximum

- The minimum button selects the time that represents the minimum value on the graph
- The maximum button selects the time that represents the maximum value on the graph
- If the minimum is selected, shift-click on the minimum button will select the time that represents the *next least* value
- If the maximum is selected, shift-click on the maximum button will select the time that represents the *next most* value

Graph Controls: Direct comparisons

- Normally, when multiple elements of a broken down statistic are selected, those elements are *stacked* on one another
- Useful for understanding how elements contribute to the whole, but less useful for directly comparing elements against one another
- The direct comparison button renders a line graph instead of a stacked graph; select multiple elements to see how they directly compare
- For non-utilization statistics, Y axis will autoscale to values of selected element(s) – beware changing Y axis!
- Clicking button again will return to stacked graph

Graph Controls: Synchronizing graphs

- When dealing with multiple graphs, it is often desirable to *synchronize* them to see correlations across data
- Click on the synchronize button to force all graphs to have the time scale of the graph on which the synchronize button was clicked
- Once synchronized, manipulating time in one graph (zooming in/out, selecting time, etc.) will have same effect on all graphs
- Individual graphs retain their sizing and any selected elements
- Once synchronized, subsequent click on synchronize button again will desynchronize graphs

Graph Controls: Drilling down

- Once an element is selected, click the drill button to *drill down* on that element in a new statistic (and new graph)
- Equivalent to right-clicking on a breakdown: brings up a menu of ways in which new statistic can be broken down
- If the current statistic is a raw statistic, menu consists of possible breakdowns for current statistic
- Shift-click on drill button for *Brendan's Rainbow*: every breakdown is highlighted as if each had been shift-clicked

Graph Controls: Saving a dataset

- The data accumulated for a statistic is a *dataset*
- By default, datasets are not saved: old data is discarded
- To save a dataset, click the save dataset button
- Once saved, data for the statistic is always gathered and saved – allowing for historical analysis
- Saved datasets can be viewed by clicking the “Saved Datasets” tab in the Analytics view, or via the CLI
- Generally needn't worry about space consumed:
  - Dataset data is highly compressible
  - Because data is stored on root device, available space is relatively ample (e.g. 500 GB in 7000 Series)

Graph Controls: Exporting to CSV

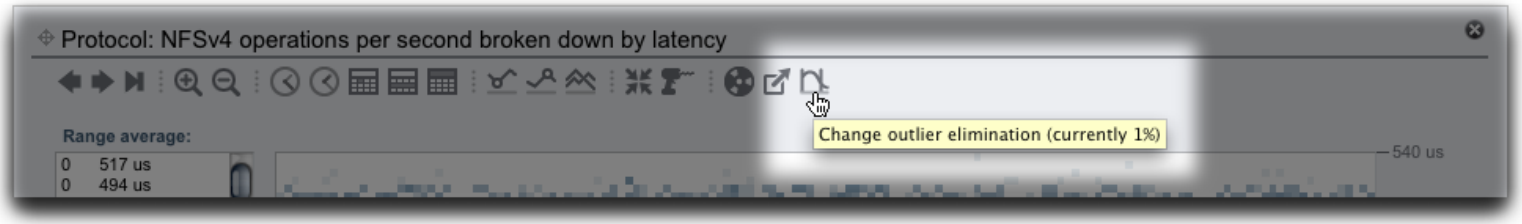
- To export to comma separated values (CSV), use the export button
- Browser will bring up a dialog box, prompting for action (save as file, open in MS Excel/StarOffice, etc.)
- Data will be at granularity of graph:
  - If zoomed in sufficiently for per-second resolution in the graph, each row will correspond to one second of data and will contain the value at that second
  - If not zoomed in sufficiently for per-second resolution, each row will contain a *range* of data, along with *minimum*, *maximum* and *average* of per-second samples

Graph Controls: Exporting to CSV, cont.

- File contains either one date/time column (if at per-second resolution) or two (if coarser than per-second)
- Date/time columns are in ISO 8601 format and in UTC
- Any selected breakdowns have their own column (if per-second resolution), or set of columns (if coarser than per-second resolution)
- Quantized breakdowns have a column for each quantization level

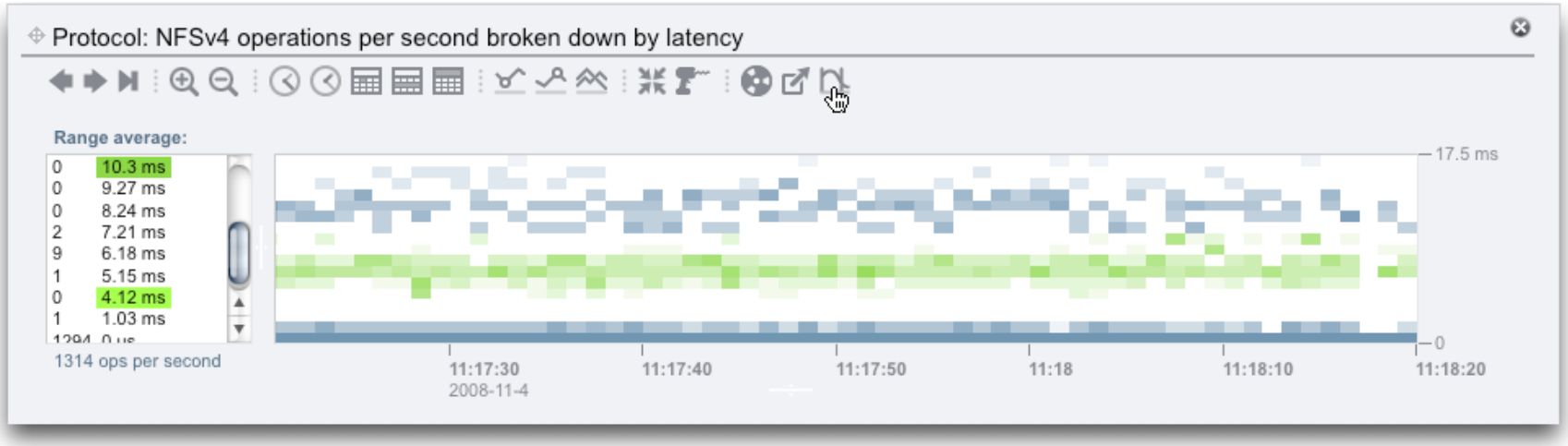
Graph Controls: Outlier elimination

- Quantized breakdowns don't (by default) show all data because doing so tends to *distort* the Y axis
- By default, *greatest 0.1%* of samples are eliminated
- This behavior can be changed via the outlier elimination button
- Clicking button will iterate through different outlier elimination percentages: 5%, 1%, 0.1%, 0.01%, 0%
- Current outlier elimination can be determined via tool tip on outlier elimination button:



Graph Controls: Outlier elimination, cont.

- Can also eliminate all samples that lie outside of a specified range
- Click on one quantization level in breakdown table, and then shift-click on another to form the range
- Shift-click on outlier elimination button to eliminate samples from graph that fall outside of selected range:

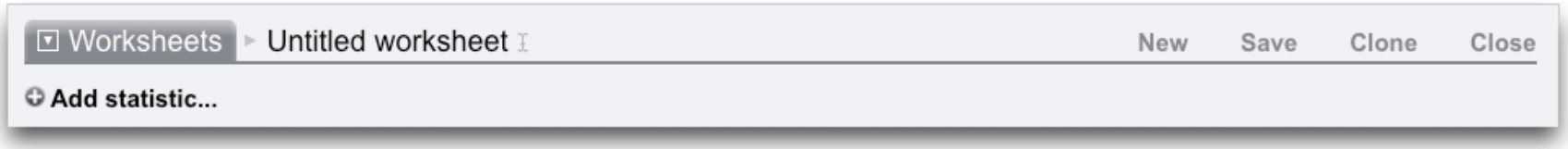


Worksheets

Saving state, encapsulating narrative

- Every open statistic is a part of a *worksheet*
- Worksheets can be saved persistently, loaded at later times or by other users
- Depending on whether or not its graphs are paused, worksheets can represent a *snapshot* of the system or a *way of understanding* the system
- Worksheet controls are along top of analytics screen:

Worksheet Controls

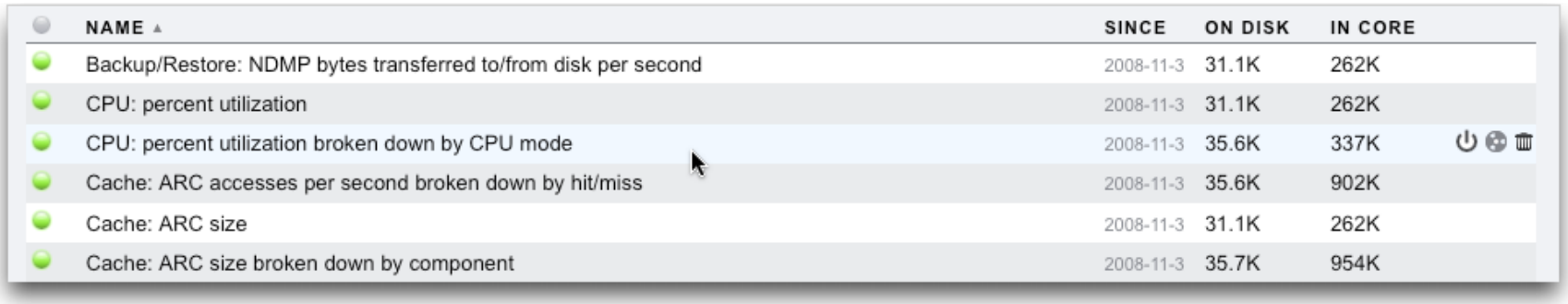


- "Worksheets" brings up a menu of open worksheets and allows the current worksheet to be changed
- Worksheet name ("Untitled worksheet" by default) can be changed by clicking on the text
- "New" creates a new worksheet
- "Save" saves current worksheet *and all datasets that it contains*
- "Clone" copies current worksheet contents into a new worksheet
- "Close" closes current worksheet
- "Add statistic..." adds statistic to current worksheet

Manipulating Datasets

Listing, suspending, resuming

- Datasets can be viewed via the “DATASETS” tab:



NAME	SINCE	ON DISK	IN CORE
Backup/Restore: NDMP bytes transferred to/from disk per second	2008-11-3	31.1K	262K
CPU: percent utilization	2008-11-3	31.1K	262K
CPU: percent utilization broken down by CPU mode	2008-11-3	35.6K	337K
Cache: ARC accesses per second broken down by hit/miss	2008-11-3	35.6K	902K
Cache: ARC size	2008-11-3	31.1K	262K
Cache: ARC size broken down by component	2008-11-3	35.7K	954K

- As with other lists in the appliance BUI, list can be sorted by a field by clicking on its header
- Light denotes *status*: on (green) is active, off (gray) is suspended
- Suspend an active dataset or resume a suspended one by clicking power icon

Manipulating Datasets

Saving, destroying, opening

- **“ON DISK” denotes *uncompressed* size – but datasets routinely get >12X compression!**
- **Destroy a saved dataset by clicking on the trash icon**
- **Save an unsaved dataset by clicking on the tape icon**
- **By default, there are quite a few saved datasets; if you destroy these, the dashboard will not contain historical information!**
- **Open a dataset in the current worksheet by clicking on its entry in the list**

Saved Worksheets

Listing, opening, appending, destroying

- Saved worksheets can be viewed via the “SAVED WORKSHEETS” tab
- A worksheet can be *opened* by clicking on its entry
- A worksheet can be *appended* to the current entry (that is, its datasets can be appended to the current worksheet) by clicking on plus icon (“+”) in entry
- A worksheet can be *destroyed* by clicking on the trash icon in its entry in the list
- By default, non-root users can see *only* the worksheets that *they* have created/saved, but worksheets can be shared via authorizations

Analytics and Thresholds

Thresholds and alert actions

- Recording information is helpful – but one may want to be notified when a statistic exceeds a specified value
- This can be done with a *threshold alert*:
  - Go to “Configuration,” select “ALERTS” and then select “Threshold alerts”
  - Click the plus (“+”) icon; all available statistics plus all saved datasets will be in a menu
  - Configure alert timing and action as desired
- Alert actions include sending e-mail, SNMP trap
- Alert actions can *also* include suspending/resuming datasets/worksheets – allowing for auto-analysis!

Analytics Authorizations

Controlling access to statistics

- Analytics allow unprecedented observability into the system – perhaps *too much* for some!
- *Any* appliance administrator can view raw statistics that do not involve any drilling down
- Authorized users can grant appliance users the authorization to drill down by *type*:
  - Go to “Configuration,” select “USERS” and then edit/create a user or a role
  - Select “Analytics” scope, then select type of breakdown and then indicate one of/both “create” or “read”
  - Click “ADD” to add authorization
- Grant powerful types (e.g., “file”) carefully!

Analytics Authorizations

Sharing worksheets

- By default, non-root users can open/modify only the worksheets that they created (root user can open/modify all worksheets)
- To give a user the authorization to see or modify a different worksheet:
  - Select “Worksheet” scope, then select Owner and select name of worksheet
  - Select one/both of “modify”/“read” and click “ADD” to add authorization
- Note that a user must *also* have the authorizations to access all statistics on a worksheet in order to open it!

Analytics via the Appliance CLI

No, no ASCII art – at least not yet

- **CLI functionality available in "`analytics`" context**
- **Manipulate datasets in "`datasets`" context:**

```
dory:> analytics
dory:analytics> datasets
dory:analytics datasets> list
DATASET     STATE   INCORE ONDISK NAME
dataset-000 active    756K  35.6K arc.accesses[hit/miss]
dataset-001 active    222K  31.1K arc.l2_accesses[hit/miss]
dataset-002 active    222K  31.1K arc.l2_size
dataset-003 active    222K  31.1K arc.size
dataset-004 active    803K  35.7K arc.size[component]
dataset-005 active    222K  31.1K cpu.utilization
```

- **"`select`" a dataset to read *n* seconds of data ("`read`") or print *n* seconds of data as CSV ("`csv`")**
- **Can also destroy/save datasets ("`destroy`", "`save`") and suspend/resume (via "`suspended`" property)**

Analytics via the Appliance CLI

Manipulating worksheets

- **Manipulate worksheets via "`worksheets`" context:**

```
dory:> analytics worksheets
dory:analytics worksheets> list
WORKSHEET       OWNER   NAME
worksheet-000   bmc     Drilling down on CIFS performance
worksheet-001   root    Running ./configure on MySQL
worksheet-002   bmc     Suboptimal performance on Wed. eve
worksheet-003   bmc     WTF is "fp-gw-08" and why is it pounding on us?
dory:analytics worksheets> select worksheet-001
dory:analytics worksheet-001> list
DATASET      DATE         SECONDS NAME
dataset-000  2008-11-3         60 nfs4.ops[client=kiowa.sf.fishpong.com][file]
dataset-001  2008-11-3         60 nfs4.ops[latency]
```

- **Individual dataset within a worksheet can be selected and read with "`csv`" command – this should be mechanism for scripting statistics queries**
- **Worksheets can be destroyed via "`destroy`" command**

Analytics via the Appliance CLI

Functionality only available via CLI

- Right now, one can only batch suspend/resume datasets via the CLI
- To suspend all datasets on the system, use "`suspend`" command from "`analytics datasets`"
- To suspend all datasets in a worksheet, use "suspend" command from within the context of the worksheet to be suspended
- Both contexts have "`resume`" equivalents

Analytics Overheads

Dataset storage

- Saved dataset data is not discarded; years after the fact per-second data is available
- This is generally not a problem: data compresses very well, and even the smallest root devices are large
- Can view the amount of total storage dedicated to analytics data by examining system storage breakdown:
  - Go to “Configuration,” select “SYSTEM”
  - Examine chart/data in left pane
  - Also available as “`configuration system`” in CLI

Analytics Overheads

Dataset storage, cont.

- If analytics data is a significant portion of total data, go to the datasets list and sort by “ON DISK” to determine large datasets
- Consider suspending data-intensive datasets, deleting useless ones
- Datasets expected to consume most on-disk data are breakdowns by file, by client, by latency and by size
- Example usage from a busy system:

Statistic	Span	Dataset Size	After Compression
CPU percent utilization	130 days	127 Mbytes	36 Mbytes
Network device bytes by device	130 days	402 Mbytes	119 Mbytes
Disk I/O operations by latency	31 days	1.46 Gbytes	515 Mbytes

Analytics Overheads

Execution overhead

- Analytics uses two different backends for its data:
  - kstat for many raw statistics and some breakdowns
  - DTrace for some raw statistics and many breakdowns (and all the interesting ones!)
- kstat-based statistics have no cost in terms of execution time – they are gathered whether datasets are enabled around them or not
- DTrace-based statistics are more invasive, and only gathered on demand – they can have an observable impact on execution time if used extensively or under high load

Analytics Overheads

Execution overhead, cont.

- Even DTrace-statistics that collect no data have a performance impact!
  - DTrace predicates still must be evaluated – and these predicates can be expensive
  - For example, “all operations from client 'kiowa'” must test client against 'kiowa' even if it never matches
- In benchmark environments or high load situations, consider suspending datasets of invasive statistics
- If uncertain, experiment with suspending all datasets via CLI and observing difference in performance

Analytics Futures

Many possibilities – big and small

- Capacity analytics for capacity planning purposes
- Displaying log events on analytics graphs
- Allowing analytics to be backed up off-appliance
- Allowing analytics to be correlated/aggregated across multiple/many appliances
- Many of the features of analytics were not conceived of in the abstract; they were explicit requests from those trying to use analytics
- If you find analytics lacking, don't be bashful – let us know!

Analytics Value

Lower $/op through smarter spend!

- Available by default in all 7000 series NAS appliances
- *There is no additional cost for analytics!*
- By allowing the user to *graphically* and in *real-time* understand appliance performance, can differentiate issues of *load* from issues of *architecture*
- Knowing the difference saves a bundle:
  - Issues of load can be *resolved* instead of throwing a useless and expensive purchase order at the problem!
  - Issues of architecture can be *resolved* by buying the right additional components (CPU, read cache, write cache, etc.), instead of wasting time and money adding more of the wrong ones!

Sun microsystems

Thank you

Learn it. Live it. Make IT real.

CEC 2008

November 9-14, 2008 Las Vegas, NV