LiveRAC: A scalable network management visualization system

Anonymous 1
Affiliation
Affiliation
author@a.com

Anonymous 2
Affiliation
Affiliation
author2@b.com

Anonymous 3
Affiliation
Affiliation
author3@b.com

ABSTRACT
We describe a scalable network management visualization system called LiveRAC that uses live streaming data and implements a novel accordion drawing + semantic zoom interaction technique. The field deployed system helps network operations staff manage complex environments. The LiveRAC visualization system shows alarm and metric data such as CPU usage and available memory for a large collection of machines simultaneously using semantic zooming, allowing the user to choose which servers to inspect with detailed charts while still showing a compressed view of the entire information space.

Author Keywords
infoviz, interfaces, accordion drawing, network management

ACM Classification Keywords
The ACM Classification keywords here.

INTRODUCTION
Network and internet services have grown enormously in the past two decades and have become critical to all aspects of modern business. Delivering these services present challenges of scale, complexity and reliability. One common approach to reliably providing network services is to contract out critical aspects of data infrastructure into highly concentrated data warehouses to be maintained by a provider which specializes in these environments. Although many technical hurdles have been overcome, there remains a human problem of monitoring and managing data warehouses which can host tens of thousands of physical and virtual servers. Network management staff must have access to specific system details while maintaining awareness of the state of the managed environment and respecting individual customer requirements.

Visualization addresses the problem of interpreting large-scale data sets by leveraging knowledge of human perception. The growing variety of network assets and the increased scale of data warehouses has resulted in a paucity of effective network management visualization tools. While an enormous number of automation, ticketing and single-system forensic applications exist to support network management, comparatively few tools offer visualization to support network operations staff and even fewer provide visualizations that are useful. Most commercially available network support tools target either high level dashboards that do little more than provide summary statistics that fail to capture the complexity of the underlying system state. For example, some dashboard systems [29, 17, 8] use thresholds that rate the monitoring state as "healthy" if some percentage of systems is available and responding. In actual data centers the environment may be very "unhealthy" if a single critical system is down, or still be healthy if five hundred machines are down for scheduled maintenance. Oversimplifications add little value for a network management professional and can even be actively harmful. Other network management tools, such as sniffers, are very effective at forensics but are difficult to apply on the scale of thousands of devices. A wide gap exists in most of these systems between overview and detail views, and this is an area which has traditionally proved fruitful for visualization. [[reference]]

This paper presents a field deployed visualization system called LiveRAC that provides both overview and detail views of live network management data and implements a novel accordion drawing + semantic zoom information visualization technique. The objective of the LiveRAC system is to provide network management staff a middle-ground between high-level overview and detail oriented tools. We begin by providing background on network management staff tasks and establishing design requirements, followed by related work. We then introduce the features of LiveRAC, and describe the system's architecture. Finally, we will discuss our formative evaluation of the system, followed by future work and conclusions.

NETWORK MANAGEMENT STAFF
The role of a network management professional in a managed hosting service environment is to meet or exceed "service level agreements" (SLAs) that have been established between the hosting provider and each of their customers. These agreements specify all aspects of the services that will be provided and typically include elements such as:

- What services will be provided to the customer
- How these services will be delivered
- How service delivery will be benchmarked
- Consequences if the service agreement is not met

These agreements often specify at very fine grained detail how long each system may be down each month, what maintenance is to be performed, and how quickly alarms must

be addressed and resolved. Adding to the complexity, customers select their own vendors for computer and networking equipment resulting in a highly heterogeneous environment with radically different configurations between customers. Delivering on the SLA agreements requires a combination of business and network management skill sets. Effective network management for thousands of devices is an activity that consists of many individual tasks. High level network management activities include:

- Alarm monitoring and response
- Incident investigation
- Capacity planning

Performing any of these high level tasks may be composed of a profusion of mid-level activities which include interpreting the state of the network environment, obtaining individual system details, reviewing log files, developing or following solution recipes, communicating information, and general problem solving.

To help manage complexity, network management staffs are divided into multiple response "tiers":

- Tier 1 is responsible for monitoring and responding to unexpected events and are typically notified of these events through an alarm & ticketing system. Tier 1 staff do initial incident investigation to see whether the problem matches a set of predefined solution recipes that have been provided for common problems. If the problem does not match a recipe, or cannot be resolved within a time window defined by the SLA, the tier 1 staff escalates the incident to tier 2.
- Tier 2 staff perform a similar function to tier 1, although they may gather more extensive system forensics and are less dependent on pre-made solution recipes. As with tier 1, if the problem cannot be resolved within the time defined by the SLA, or if the solution requires architectural changes to the monitored systems it will be passed on to tier 3.
- Tier 3 (and higher) network operations staff have a less reactive role than in tier 1 and 2. In addition to resolving problems passed on from tier 2, they are often engaged in a capacity planning and architectural roles.

We have focused on developing our system for a group of tier 3 professionals called 'Life Cycle Engineers' (LCEs). LCEs are senior network operations staff who are assigned specifically to one or more customer accounts. Key aspects of their role include understanding the customer's environment as it is configured in the data warehouse, assessing and analyzing current system state, forecasting future requirements and communicating this information with customers.

NETWORK VISUALIZATION SYSTEM DESIGN REQUIREMENTS

Our initial requirements for this project were generated from the first author's professional background in deploying network management and monitoring systems, performing network forensics and architecting data centre solutions on the scale of thousands of CPUs. We validated and augmented these requirements through collaboration and iterative design with senior network management professionals and researchers at AT&T Inc., one of the largest managed hosting services providers in the world.

High level requirements include allowing a network professional to:

- *Assimilate overall system state at a finer level of granularity than simple dashboards of how many systems were "up" or "down".* The data presented must support the network professional's own analytical reasoning.
- *Obtain monitored system metrics without sacrificing overview.* Typically navigating to view individual metrics for a system means sacrificing overview of the system state. Multiple monitors helps, but requiring head movement or making large eye saccades makes it difficult to perform direct comparisons and maintain complete system awareness.
- *Correlate alarms and tickets with system metrics.* Most automated systems suffer from huge numbers of false positives. Many of these are handled by automated filters, but some require investigation by an ops staff member. System metrics provide a good second-level filter for determining if an alarm is a false positive or if further investigation is required. False positives from a single data warehouse can be on the order of hundreds of thousands of alarms per month.
- *Dynamically customize thresholds.* Operations staffs need to be able to fine tune threshold parameters of the visualization to meet SLA requirements of particular customers. Making this dynamic allows easier data exploration by letting staff members ask questions such as, "What machines have CPU peaks of more than 85%?" and obtain answers quickly.
- *View and export raw data.* The raw stat data and full alarm text is required for investigation of specific incidents. To better integrate with existing tools, and to share data with customers, the visualization system needs the capability to export the raw data.

RELATED WORK

Information Visualization

Information visualization leverages the data collection and aggregation power of computers to augment cognition. The human visual channel is the highest bandwidth perceptual system we possess for information acquisition [28]. Information visualization systems sort and display information in a manner that allows an analyst to create mental models of the underlying data, and mine this data for correlations and outliers.

Focus+context [38] is a family of visualization techniques which provide context-situated data views for user-selected

regions of interest. These techniques provide useful contextual data in support of the user's primary activity and assist user navigation within the dataset by providing spatial position data. Many of these techniques use distortion or aggregation to concurrently view the entire data set [14, 10, 11, 33, 7]. Distortion-free approaches to focus+context displays also exist using glyphs [9, 31] or rectilinear approaches as in Table Lens [32]. LiveRAC shares the TableLens rectilinear focus+context approach to visualizing tabular data, but uses a different interaction approach called accordion drawing (AD) introduced below. Unlike LiveRAC, TableLens does not support live, streaming data or dynamic modification of the data set.

LiveRAC uses a visualization and interaction technique called accordion drawing (AD) [18]. Accordion drawing combines rubber sheet navigation [34] and guaranteed visibility (GV). Rubber sheet navigation is a focus+context interaction metaphor where the user manipulates the display as though it was a rubber sheet tacked down at the borders. Navigation operations can stretch and compress arbitrary regions of the display. Multiple focus areas are possible with rubber sheet navigation, as different regions can be stretched to a desired size. During navigation landmarks or regions of interest may become compressed such that they are not visible if another region of the display is dramatically expanded, or when viewing a very large data set. In many application domains it can be desirable to have regions of the display which are always visible irrespective of navigation activities. These regions, called critical zones [20], are provided by GV. Visualization systems implementing GV ensure that marked regions will remain visible regardless of the information density. LiveRAC uses and extends the PRISAD infrastructure by introducing fully dynamic data structures that allow grid lines to be added and removed at run time, and providing a framework for semantic zoom in AD.

Semantic zoom is a visualization technique that represents graphic objects differently depending on the apparent size of the object to the user. This technique does not simply increase the polygon count or detail of an object that is occupying more display pixels, but changes the type of visual representation to one most appropriate for its display area. For example, when occupying a small region of space a calendar object might display only a list of high priority events and the dates of those events in text. When the region is enlarged, the calendar can lay out the more familiar month view. The first visualization system to implement semantic zoom was Pad [30], and it has also been integrated into focus+context visualizations [4]. LiveRAC is the first system to implement a semantic zoom framework in accordion drawing.

Matrix layouts are a common way to encode tabular data in information visualization systems. [3, 32, 36, 37, 39] With computer supported visualization, it is possible to interactively reorder a matrix view to help find correlations in the data, an idea documented by Bertin [5], and also applied by several visualization systems [39, 15, 21, 22]. LiveRAC uses a reorderable matrix layout for data. The key differences between LiveRAC and previous approaches are the use of the accordion drawing metaphor. LiveRAC is the first accordion drawing visualization system to support dynamically adding, removing and reordering data.

Statistical Graphics

Statistical graphics, also called data graphics or quantitative graphics, is the projection of abstract shapes representing observed quantitative data onto a co-ordinate system. Statistical graphics are used extensively in scientific and business literature, and have been studied by both the statistics and design communities. [12, 6, 41, 42, 43, 44]. LiveRAC's graphics library includes sparklines, line charts, scatter plots, bar charts and histograms.

Time Series Data

Time series data consists of any data elements which have a time dimension, such as an ordered list of [time, value] pairs. Data from most sources which are sampled at regular intervals preserve a time dimension, including many science and engineering data sets. Other visualization systems which have used time series data includes [48, 47, 16, 26]. All data visualized by LiveRAC has a time dimension. LiveRAC provides a matrix view of time series data. LiveRAC also uses dynamic queries [35] in a linked view for interacting with the time dimension, similar to TimeSearcher [16]. LiveRAC differs from TimeSearcher in application domain, and the use of the accordion drawing visual metaphor.

Network and Systems Visualization Tools

A large body of visualization tools have been developed for performing security and fault analysis on computers and computer networks. These range from basic tools such as packet analysis software like Ethereal [13] to web based viewers for intrusion detection alarm data such as Analysis Console for Intrusion Databases (ACID) [2]. A plethora of interactive visualization tools have been developed for studying this data, including [25, 3, 23, 1]. SWIFT [24] is a system developed at AT&T Labs, Inc. - Research to collect and display streaming network data. LiveRAC connects to the SWIFT system, which is described in more detail in Section .

Interaction technique papers

* Need some interaction technique, participatory design & field study related work references here * Some possibilities: [27, 50, 49, 46]?

THE LIVERAC APPLICATION

LiveRAC is a visualization system for monitoring and exploring alarm and statistical data from network assets. The main window consists of a large data-view area, and a tool panel located immediately below. Unless otherwise specified, the discussion below refers to the data-view region.

Layout

LiveRAC presents a matrix view of the data in an accordion drawing infrastructure. Rows represent network devices, and each column represents a group of one or more

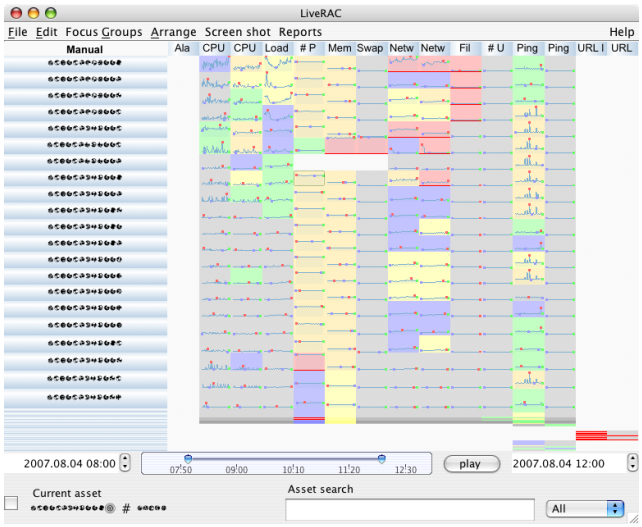


Figure 1. LiveRAC is a network management visualization system. Rows represent network assets, columns represent groups of monitored parameters. Users interact with the application by 'stretching' and 'squishing' parts of the data area. Regions-of-interest display additional data, compressed regions show high-level aggregate information.

monitored parameters. Each matrix cell contains an area-aware data graphic representation of the underlying data. The number of nodes in the data set can be larger than the number of display pixels available. Regions with data density that exceed the number of pixels in the display are aggregated. The user specifies foci by stretching and squishing display regions through navigation actions on the data set.

Color

The smallest data graphic representation for a single cell is a colored box. This color is preserved as a background even when other data graphics are drawn overtop at higher levels of zoom. Colors can be user defined, although by default LiveRAC uses the color profile adopted for another internal network management tool used at AT&T. Although these are not maximally discriminable colors, we have found they are adequate in practice and preserving familiarity for end users was a design priority.

Color saturation is used to encode data density. A non-aggregated cell has a base saturation of 25%. As a cell is expanded, the saturation level decreases proportionally with area to a minimum of .05%. [[Magic numbers may change without notice. Void where prohibited. Consult your doctor before entering these values.]] We follow the color use guidelines of Ware [45] recommending that large areas should use de-saturated colors. Decreasing saturation level also makes the semantic representation contained within the cell more readable by providing increased contrast. The enforced minimum ensures the severity level of the cell will always be visible regardless of how large it is expanded. Our formula for computing cell saturation is provided in Equation 1, where we define B as being the base saturation of a non-aggregated cell, M as being the minimum saturation for a cell, K as the size at which minimum saturation is achieved and S as the current size of the cell.

$$\frac{B-M}{K}S \tag{1}$$

Saturation increases when cells are aggregated as described below.

Aggregation

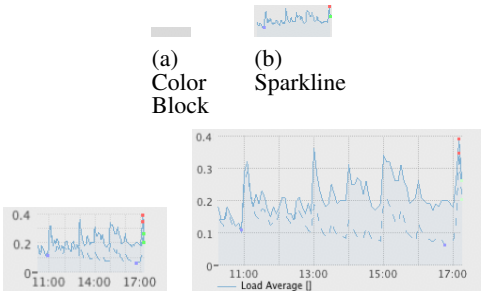
Data aggregation is applied when there are more devices in a region of the display than pixels available to display them. Aggregation allows the system to take a rules-based approach for selecting what value to display for a given pixel. The naive alternative is to draw every value underneath the pixel, with the first, last, or blended value being the one the user sees. In addition to the obvious inefficiency of this approach, the result is unlikely to yield the most relevant information for the user. LiveRAC provides four aggregation functions: minimum, maximum, mean and cardinality.

Guaranteed visibility

Critical alarms, critical threshold incursions and search results are marked using guaranteed visibility (GV) groups. The critical markings were selected on the basis of their importance to users of the system. We had been told explicitly by senior network management staff, even those that are not involved in day-to-day operations, that they like to check in and find out how many critical events have occurred. GV ensures these critical alarms are visible.

Search results must also be marked with GV. Without using GV, search results might be hidden in highly aggregated regions.

Semantic Zoom



(a) Color Block (b) Sparkline

(c) Reduced line chart (d) Full size line chart

Figure 2. An example of semantic zoom on a line chart. (a) Only a color swatch is displayed. (b) The chart has been changed to a sparkline graphic. (c) The line chart has been reduced in size, the legend has been removed and labels are more sparse. (d) A full sized line chart is displayed.

Semantic zoom (SZ) is a visualization technique where different representations of data are presented at different levels of zoom. LiveRAC uses SZ to provide a hierarchy of representations for data. We designed a GUI configuration panel to assist users in creating this hierarchy. Representation specific options are available through the GUI such as

chart grids, labeling, borders and padding. An example hierarchy of representations is displayed in Figure . We developed a library called jGLChartUtil to provide data graphic representations.

Interaction

LiveRAC uses the visualization and interaction technique called accordion drawing (AD) [18] described in the related work section. LiveRAC users select a rectangular region using the mouse, and then use click-drag operations to stretch and squish the selection. The edges of the display are fixed such that all data is visible at all times on the display, although some data may be in a highly aggregated state. All navigation actions are animated which helps prevent the user from becoming lost [[ref]].

LiveRAC provides keyboard and mouse shortcuts to help users navigate the data more quickly. These rapid navigation functions include: zooming for a single cell, grow an individual column, direct manipulation of grid lines, grow a device group, and zoom out to overview. We also introduce user defined groups of network devices and columns called focus groups. Focus groups can be grown or shrunk together. Items added to a focus group do not have to be adjacent. This lets the user grow multiple focus areas quickly and simultaneously if they have groupings of recurring interest. LiveRAC is the first accordion drawing system to introduce focus groups.

Temporal navigation

LiveRAC lets users look at a window of time which can scale from minutes to months. Users can select the time window using a double edged slider. LiveRAC also supports VCR-like control plays forward through data in realtime or faster than realtime. Real time playback allows users to view live data, faster than realtime playback is useful for rapidly viewing archived data. LiveRAC is the first accordion drawing system to support a time dimension.

Linking

Users need to correlate events visible in data graphic representations between network devices and columns. Because the size of focus regions in the matrix view may be different, and some graphic representations may be too small to provide labels, it can be difficult for a user to make spatial judgments of precise time intervals between columns. A linked mark is therefore provided. When the user moves the mouse over one graphic representation, the same instant in time is marked on all graphics.

Reordering and Sorting

Arranging and grouping rows and columns in the matrix view is vital to making small-multiples comparisons needed for exploratory data analysis [5]. Users can specify an arbitrary ordering of rows and columns in LiveRAC. Because the number of monitored network devices is typically quite large, assets can be sorted using metadata such as asset name, location, logical group or customer. Users can also sort devices based on column data. For example, systems with the highest load average can be sorted to the top or bottom.

Search

LiveRAC provides a progressive search mechanism for network device names and metadata. A keyboard shortcut supports growing all search results simultaneously.

ARCHITECTURE

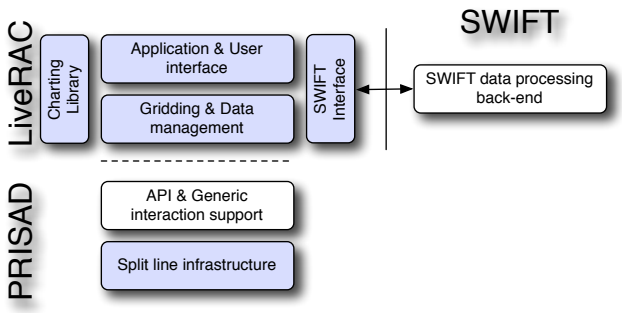


Figure 3. LiveRAC uses a modular architecture. The PRISAD accordion drawing infrastructure provides API's on which the LiveRAC application is built. The SWIFT interface layer provides a network client-server interface to the SWIFT data collection and processing services. Shaded boxes indicate modules where we have made significant contributions.

LiveRAC implements a threaded, modular architecture that utilizes the PRISAD [18] accordion drawing API. The application layer manages user interaction, enqueues data requests and renders the data view. The gridding and data management layer maintains the data structures necessary for addressing each cell in the matrix layout, stores data retrieved from the server and maintains information regarding cell representations. PRISAD provides an API for accordion drawing. Extensive modifications were made to the PRISAD grid infrastructure, described in Section to provide the dynamic add, remove and reorder functionality required by LiveRAC.

LiveRAC's implementation is based on Java, with native bindings to hardware accelerated OpenGL through the JOGL library [19]. We designed LiveRAC to use a client-server model for connecting to a data server. Currently LiveRAC has bindings for an AT&T internal data server called SWIFT which collects and pre-process data, but other bindings will be developed in the future. The client-server architecture enables the visualization client to run on a standard desktop PC while supporting interaction with gigabytes of data. The server can collect data on an ongoing 24/7 basis and perform data processing operations such as computing rolling, weighted averages and aggregates.

SWIFT

SWIFT [24] is a set of data storage, aggregation and visualization tools that allows data from many distinct sources to be integrated into a single self-describing data format. Data sources currently integrated into SWIFT include SNMP, intrusion detection systems and Microsoft Windows system monitors. The underlying system is extensible so virtually any data source can be mapped into the schema.

SWIFT queries and filters can be compiled into shared libraries that execute rapidly over large sets of records. Several visualizations have been developed for SWIFT, including: geographic, node-link, and table based.

LiveRAC provides a new visualization front-end for SWIFT data. Where previous SWIFT visualizations have focused on the physical relationships between network devices and high-level dashboards, LiveRAC's goal is to provide a scalable, highly-interactive, information-dense matrix encoding for the time-series data associated with network devices. LiveRAC queries SWIFT progressively when users select focus regions. Initially, only overview data is loaded. Data required for plotting graphic representations is queried on an on-demand basis. This keeps the memory requirements for the client modest despite the volume of data that can be viewed.

Data model abstraction & Configuration

Although we use network management terminology throughout this paper to describe LiveRAC features and visual encoding, LiveRAC has been designed with abstraction as a core design principal. These abstractions include:

- Network assets can be replaced with any type of data *source*.
- Columns are actually *strips* of information that need not be confined to a vertical orientation in the future.
- Parameters being monitored from network assets are simply *input channels*.
- Many *strips* used similar data types that would use the same set of graphical representations to display at each level-of-detail. These sets of graphical representations could be abstracted and shared as *visual templates*.
- Charts are graphical data representations, other non-chart representations can be used in cells so long as they can be stored in an OpenGL display list.

Dynamic accordion drawing

To fully support scalable dynamic reordering, adding, and removing of rows & columns, the following requirements were established for each grid line:

1. Worst case logarithmic insert and remove operation
2. Linear scalability in memory usage
3. Arbitrary ordering
4. Locate, insert and remove grid lines by index number in log time without reindexing after insert / remove / reorder operations

We used a red-black tree[40] as the basis for our dynamic data structure which supported requirements 1 - 3. However, traditional tree structures do not support requirement 4, necessitating the development of a custom red-black tree implementation. LiveRAC is the first accordion drawing based system to support this fully dynamic, scalable grid system. The infrastructure can support over a million grid lines on each axis.

FORMATIVE EVALUATION

We designed the LiveRAC visualization system using an ongoing iterative approach that involved research and network professionals from AT&T from the proof-of-concept through to the high-fidelity prototype stages. Data collected for our design includes:

- Group meeting notes
- Interaction log data
- Surveys
- Individual debriefs

We will discuss a set of visual case studies that provide working examples of findings from real data. We will then discuss the preliminary results from our field deployment of LiveRAC, and then summarize our key findings.

Visual case studies

Visual case studies in Figure 4, Figure 5 and Figure 6 demonstrate results from using the LiveRAC system. We compare performing certain types of queries using LiveRAC to other web-based visualizations available for SWIFT.

In Figure 4 obtaining the same information using the tabular and chart data in the SWIFT visualizations would require at least four display windows to be open just to display the data concerning the four most heavily loaded devices. Furthermore, each of these display windows would require scrolling to view charts. Obtaining the list of most heavily loaded devices would require writing a custom query and report, or navigating to each asset manually.

The erroneous data in Figure 5 would only be detected using other SWIFT visualizations by navigating specifically to the charts page for the devices with the malfunction monitoring daemons. The red swatches in LiveRAC drew immediate attention, and the wild variations in memory use are clearly visible as being unusual even in the no-label sparkline view.

The details available in Figure 6 would be almost impossible to derive with other SWIFT visualizations. We can see sparkline data for 20 devices in this view, and more aggregated information for half a dozen others. Comparing the alarm event and checking the high-water mark on the other devices would require 20 windows open to do side-by-side comparisons, plus the navigation actions required to load them. Creating this view in LiveRAC required two rectangle-select-drag operations that were completed in about 20 seconds.

All of the material for the visual case studies was found through interpreting the overview and zooming on regions of interest. Even without selecting focus regions it is possible to see important details that would be hidden by a dashboard approach. For example, it is trivial to see which critical thresholds have been exceeded and what asset groups were most severely affected, even when viewing 4000 assets simultaneously.

Figure 4. Around 4000 network assets are sorted by their mean load average. The topmost region showing the most heavily loaded assets has been slightly expanded. Four devices have exceeded the "Major" threshold (orange), but none exceed the critical (red) threshold for load. Furthermore, we can see that although high load average systems tend to have the highest network utilization (marked critical), they are configured with sufficient memory as indicated by low swap utilization (gray).

Although LiveRAC found results for these queries that require significantly more navigations in other SWIFT visualizations, the SWIFT visualizations offer types of views that are not available through LiveRAC. LiveRAC was not designed to support the geographic, block-diagram and device inventory views provided by the other SWIFT visualizations. LiveRAC enhances the SWIFT visualization suite by supporting queries that were not previously possible.

Participant feedback and Discussion

We provided LiveRAC to a limited group of senior network management operations staff. At the time of writing thirteen AT&T internal participants had tried the LiveRAC system, with four of these being regular users. Our approach has been to identify and collaborate with participants who were most interested in trying experimental new technology. Of the participants who tried the system but are not frequent users, most are management and do not regularly monitor system state as part of their work activities. Their interaction with LiveRAC was mostly based on curiosity after seeing demos or finding out about the system from email communications. The four regular users of our system were all part of the life cycle engineer team.

We found that LiveRAC fit well into the middle ground of providing more information than dashboard approaches, while still providing details on demand. This was supported by feedback from users in multiple discussions. For example, in one discussion Bill, a principal systems architect, told us, "What I liked about the, um, LiveRAC, ... where, if there was a particular alarm, you could put that, uh, vertical line and look at all the other parameters, where like if CPU spiked ... you could look at all the other parameters and see where they are, or what we found was there was critical alarms on, uhh, ping test, but when you looked at the cpu utilization was extremely low, ... so, you, you could get a sense of the health of the asset... ". He was also impressed by the overview capabilities: "With LiveRAC we can see all the assets for a customer, 4000 assets on the screen."

We discussed the possibility of creating a "health value" metric for each device based on weighted analysis of its parameters with members of the life cycle engineering team and their managers. However, consensus was that even should such a metric be possible, a tool would still be needed to investigate the validity of any such number. The complexity of the environment tends to cause such highly simplified values to yield too many false positives or false negatives to be trusted by end users. Support from visualization systems like LiveRAC allow users to arrive at conclusions they can trust because they can see the details that were used to derive the result. [[I could add trust questions to an exit survey to better support.]]

We were interested in what other commercial tools had been used in the network operation data centers to manage sys-

Figure 5. High used memory levels tripped the critical threshold values (red) for these two systems. Zooming in on the two flagged systems, the author discovered the SWIFT data collection system was returning erroneous negative memory values. This was reported to the data collection team, who were able to fix the problem the day before before they received complaints from members of the network operations team that some of the systems were reporting strange memory values.

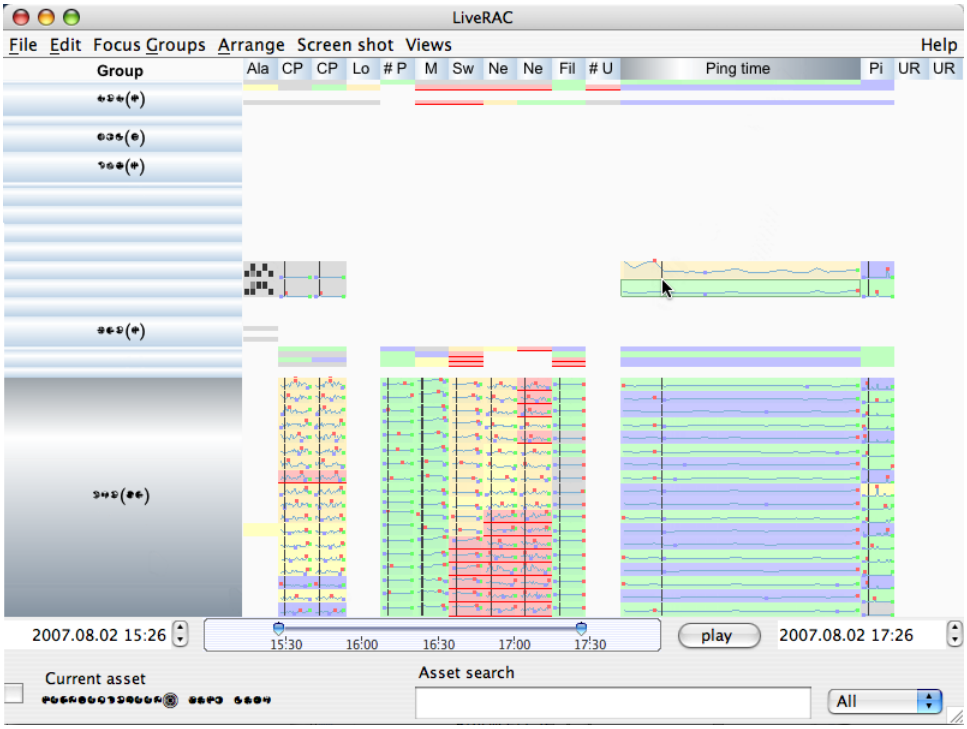


Figure 6. An network ops investigator can respond to the ping latency alarm by enlarging the flagged system, requiring only a move of the mouse and two clicks forward on the scroll wheel. Not only has an alarm been generated but the latency exceeded the threshold and has been flagged to orange. We can quickly compare the latency to other systems and see this asset is experiencing significantly longer ping delays, although we can also see that the high water mark is similar across a large number of assets indicating a period of intense activity.

tems and how these tools had performed. One reason operations adopted SWIFT is it provided capabilities not available in the other systems they had tried. Tim, a senior technical architect, described the commercial enterprise system they had been using previously as "totally inadequate". LiveRAC's unique AD+SZ visualization approach adds capabilities not available in other systems.

In two separate group discussion sessions, participants stressed the importance of being able to produce reports that integrated into their tool chain. This was a critical function as a key component of the LCE role requires interfacing and data sharing with customers. To support this requirement, we developed a Microsoft Excel export reporting system. We believe integration into the end user tool chain is a vital, and often neglected, component of visualization systems.

Challenges

Visualization systems have not had significant penetration into corporate environments and many barriers to adoption exist. Some of these are common to the introduction of any new, complex system. However, this barrier is higher for visualization research since it tends to employ novel data representations and interaction techniques. We believe that these systems can be shown to have value, but expect that adoption will be a slow and time-consuming process even with strong research results.

Overcoming training barriers has been a significant challenge in our field deployment. The situation is exacerbated by time constraints of our participants and their disparate physical locations, problems which would be faced introducing any new visualization system in a corporate environment. We provided written documentation via the same internal web site used to launch the client, but we were not surprised to discover that very few of our participants read it. A screencast video was available at the same internal site as the documentation and could be watched as often as the users desired. [[web server logs for # of views]] Although the screencast helped, the interaction method is sufficiently different from interfaces participants and the time our participants have available to apply to learning the system is such that we feel it may be several months before it is integrated into their work flow.

FUTURE DIRECTIONS

The encouraging preliminary results from participants in the project and the results we have obtained by interacting with the system have led us to believe we should bring more participants into the project and develop a full-scale field study. We believe the LiveRAC visualization system may be helpful to network management staff at response tier 2, and other areas of tier 3. There will need to be continued development of the LiveRAC visualization system to improve usability and provide features that will better integrate the system into the network management staff tool chain. Our objective with the field study is to understand where LiveRAC evolves into their workflow and what kinds of discoveries they are able to make with it.

Some of the lower level objectives of the field study will include:

- What types of queries will users use LiveRAC for?
- Do users use multiple-focii?
- Which interaction methods do the users prefer when using AD?

The results of our formative evaluation have identified a number of features we plan to implement in the next version of LiveRAC. These include:

- More compact heatmap based representations
- Improved query performance
- Visual representations of server-based feature detection
- Realtime updates for all column group and template modifications

We also plan to apply the LiveRAC system to other data back-ends and develop additional data abstractions to make it a more flexible, general visualization platform.

CONCLUSION

LiveRAC is an effective, field deployed visualization system for network management data. LiveRAC supports network operations staff tasks by targeting the middle ground between high level dashboards and low level network tools. LiveRAC's combined accordion drawing and semantic zoom allows a system overview to be present while showing varying levels of detail in user-specified regions of interest. LiveRAC's reorderable matrix visual encoding and high data density allows small multiples comparisons between data graphic representations.

Our explorations with the LiveRAC tool and the feedback from users of the system suggest that LiveRAC is effective in a field environment. Participants were particularly enthused and impressed with the capability to see meaningful data about 4000 assets simultaneously. As a flexible system with several layers of abstraction, LiveRAC will be applied to more data sets in the future.

ADDITIONAL AUTHORS

REFERENCES

1. Kulsoom Abdullah, Chris Lee, Gregory Conti, John A. Copeland, and John Stasko. IDS RainStorm: Visualizing ids alarms. In *VIZSEC '05: Proceedings of the IEEE Workshops on Visualization for Computer Security*, page 1, Washington, DC, USA, 2005. IEEE Computer Society.
2. Analysis Console for Intrusion Databases (ACID). downloadable at: http://acidlab.sourceforge.net/, cited July 26, 2006.
3. Robert Ball, Glenn A. Fink, and Chris North. Home-centric visualization of network traffic for security administration. In *VizSEC/DMSEC '04:*

Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security, pages 55–64. ACM Press, 2004.

4. Lyn Bartram, Albert Ho, John Dill, and Frank Henigman. The continuous zoom: a constrained fisheye technique for viewing and navigating large information spaces. In *UIST '95: Proceedings of the 8th annual ACM symposium on User interface and software technology*, pages 207–215, New York, NY, USA, 1995. ACM Press.

5. J. Bertin. *Graphics and graphic information processing*. Walter de Gruyter, Berlin, Germany, 1981.

6. J. Bertin. *Semiology of Graphics*. University of Wisconsin Press, 1983.

7. Eric A. Bier, Maureen C. Stone, Ken Pier, William Buxton, and Tony D. DeRose. Toolglass and magic lenses: the see-through interface. In *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 73–80, New York, NY, USA, 1993. ACM Press.

8. BMC Patrol. http://www.bmc.com/, cited 26 Aug 2007.

9. Stuart Card and David Nation. Degree-of-interest trees: A component of an attention-reactive user interface. In *Proceedings of Advanced Visual Interface*, pages 231–246, 2002.

10. M. Sheelagh T. Carpendale, David J. Cowperthwaite, and F. David Fracchia. 3-dimensional pliable surfaces: for the effective presentation of visual information. In *UIST '95: Proceedings of the 8th annual ACM symposium on User interface and software technology*, pages 217–226, New York, NY, USA, 1995. ACM Press.

11. Sheelagh Carpendale, John Ligh, and Eric Pattison. Achieving higher magnification in context. In *UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 71–80, New York, NY, USA, 2004. ACM Press.

12. William S. Cleveland. *The elements of graphing data*. Wadsworth Publ. Co., Belmont, CA, USA, 1985.

13. Gerald Combs. Ethereal. downloadable at: http://www.ethereal.com/, cited July 5, 2006.

14. G. W. Furnas. Generalized fisheye views. In *CHI '86: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 16–23, New York, NY, USA, 1986. ACM Press.

15. Martin Graham and Jessie Kennedy. Exploring and examining assessment data via a matrix visualisation. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pages 158–162, New York, NY, USA, 2004. ACM Press.

16. Harry Hochheiser and Ben Shneiderman. Interactive exploration of time series data. In *DS '01: Proceedings of the 4th International Conference on Discovery Science*, pages 441–446, London, UK, 2001. Springer-Verlag.

17. HP Inc. HP OpenView. http://www.managementsoftware.hp.com/, cited July 5, 2006.

18. James Slack and Kristian Hildebrand and Tamara Munzner. PRISAD: A partitioned rendering infrastructure for scalable accordion drawing. In *INFOVIS '05: Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization*, page 6, Washington, DC, USA, 2005. IEEE Computer Society.

19. The JOGL API Project. https://jogl.dev.java.net/, cited August 6, 2006.

20. Susanne Jul and George W. Furnas. Critical zones in desert fog: aids to multiscale navigation. In *UIST '98: Proceedings of the 11th annual ACM symposium on User interface software and technology*, pages 97–106, New York, NY, USA, 1998. ACM Press.

21. Robert Kincaid. VistaClara: an interactive visualization for exploratory analysis of DNA microarrays. In *SAC '04: Proceedings of the 2004 ACM symposium on Applied computing*, pages 167–174, New York, NY, USA, 2004. ACM Press.

22. Robert Kincaid and Heidi Lam. Line graph explorer: scalable display of line graphs using focus+context. In *AVI '06: Proceedings of the working conference on Advanced visual interfaces*, pages 404–411, New York, NY, USA, 2006. ACM Press.

23. Hideki Koike and Kazuhiro Ohno. Snortview: visualization system of snort logs. In *VizSEC/DMSEC '04: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, pages 143–147, New York, NY, USA, 2004. ACM Press.

24. Eleftherios E. Koutsofios, Stephen C. North, Russell Truscott, and Daniel A. Keim. Visualizing large-scale telecommunication networks and services (case study). In *VIS '99: Proceedings of the conference on Visualization '99*, pages 457–461, Los Alamitos, CA, USA, 1999. IEEE Computer Society Press.

25. Kiran Lakkaraju, Ratna Bearavolu, Adam Slagell, William Yurcik, and Stephen North. Closing-the-loop in NVisionIP: Integrating discovery and search in security visualizations. In *VIZSEC '05: Proceedings of the IEEE Workshops on Visualization for Computer Security (VizSec'05)*, page 9, Washington, DC, USA, 2005. IEEE Computer Society.

26. Jessica Lin, Eamonn Keogh, Stefano Lonardi, Jeffrey P. Lankford, and Donna M. Nystrom. Visually mining and monitoring massive time series. In *KDD '04:*

Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 460–469, New York, NY, USA, 2004. ACM Press.

27. Carman Neustaedter and A. J. Bernheim Brush. "linc-ing" the family: the participatory design of an inkable family calendar. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 141–150, New York, NY, USA, 2006. ACM Press.
28. Tor Norretranders. *The User Illusion*. Viking, 1999.
29. OpenNMS. http://www.opennms.org/, cited July 5, 2006.
30. Ken Perlin and David Fox. Pad: an alternative approach to the computer interface. In *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 57–64, New York, NY, USA, 1993. ACM Press.
31. Catherine Plaisant, Jesse Grosjean, and Benjamin B. Bederson. SpaceTree: Supporting exploration in large node link tree, design evolution and empirical evaluation. In *INFOVIS '02: Proceedings of the IEEE Symposium on Information Visualization (InfoVis'02)*, page 57, Washington, DC, USA, 2002. IEEE Computer Society.
32. Ramana Rao and Stuart K. Card. The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In *CHI '94: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 318–322, New York, NY, USA, 1994. ACM Press.
33. George G. Robertson and Jock D. Mackinlay. The document lens. In *UIST '93: Proceedings of the 6th annual ACM symposium on User interface software and technology*, pages 101–108, New York, NY, USA, 1993. ACM Press.
34. Manojit Sarkar, Scott S. Snibbe, Oren J. Tversky, and Steven P. Reiss. Stretching the rubber sheet: a metaphor for viewing large layouts on small screens. In *UIST '93: Proceedings of the 6th annual ACM symposium on User interface software and technology*, pages 81–91, New York, NY, USA, 1993. ACM Press.
35. Ben Schneiderman. Dynamic queries for visual information seeking. *IEEE Softw.*, 11(6):70–77, 1994.
36. Harri Siirtola. Interaction with the reorderable matrix. In *IV '99: Proceedings of the 1999 International Conference on Information Visualisation*, page 272, Washington, DC, USA, 1999. IEEE Computer Society.
37. James Slack, Kristian Hildebrand, Tamara Munzner, and Katherine St. John. Sequencejuxtaposer: Fluid navigation for large-scale sequence comparison in context. In *German Conference in Bioinfromatics*, 2004.
38. Robert Spence and Mark Apperley. Data base navigation: an office environment for the professional. *Readings in information visualization: using vision to think*, pages 333–340, 1999.
39. Chris Stolte, Diane Tang, and Pat Hanrahan. Query, analysis, and visualization of hierarchically structured data using polaris. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 112–122, New York, NY, USA, 2002. ACM Press.
40. Thomas H. Cormen and Charles E. Lieserson and Ronald L. Rivest. *Introduction to Algorithms*. MIT Press, 1990.
41. Edward Tufte. *The Visual Display of Quantitative Information*. Graphics Press, first edition, 1981.
42. Edward Tufte. *Envisioning Information*. Graphics Press, 1990.
43. Edward Tufte. *Beautiful Evidence*. Graphics Press, first edition, 2006.
44. Edward R. Tufte. *Visual Explanations: Images and Quantities, Evidence and Narrative*. Graphics Press, February 1997.
45. Colin Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers, second edition, 2004.
46. Martin Wattenberg. Visual exploration of multivariate graphs. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 811–819, New York, NY, USA, 2006. ACM Press.
47. Marc Weber, Marc Alexa, and Wolfgang Műller. Visualizing time-series on spirals. In *INFOVIS '01: Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01)*, page 7, Washington, DC, USA, 2001. IEEE Computer Society.
48. Jarke J. Van Wijk and Edward R. Van Selow. Cluster and calendar based visualization of time series data. In *INFOVIS '99: Proceedings of the 1999 IEEE Symposium on Information Visualization*, page 4, Washington, DC, USA, 1999. IEEE Computer Society.
49. Mike Wu, Ron Baecker, and Brian Richards. Participatory design of an orientation aid for amnesics. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 511–520, New York, NY, USA, 2005. ACM Press.
50. Ron Yeh, Chunyuan Liao, Scott Klemmer, François Guimbretière, Brian Lee, Boyko Kakaradov, Jeannie Stamberger, and Andreas Paepcke. Butterflynet: a mobile capture and access system for field biology research. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 571–580, New York, NY, USA, 2006. ACM Press.