Interactive Visual Exploration of Service Level Objectives

Authors
Kim, Han Suk
Verma, Sunita
Wilkes, John

Publication Date
2011-05-02

Interactive Visual Exploration of Service Level Objectives

Han Suk Kim[†1], Sunita Verma[‡2], and John Wilkes[§2]

[1]University of California San Diego, USA
[2]Google Inc., USA

Abstract
Service Level Objectives (SLOs) describe the desired behavior of a service. They present agreed-upon target behaviors (e.g. capacity, performance, availability, cost) that the clients of a service can expect, and that the service provider can manage against. Both clients and service providers need a way to monitor how a service is doing against its SLOs, and display the result of that monitoring in an intuitive fashion.
This paper describes a tool that provides interactive visual exploration for services; makes it easy to navigate around the multi-dimensional space of services, their instances, users, and metrics; and is driven by the machine-readable description of the SLO itself, so that changes can be propagated automatically. The contributions of the paper include a description of the requirements and implementation of our system; the techniques used to display aspects of — and navigate around — the multi-dimensional space; and how the SLOs are used to automate the display of service-specific information.

Categories and Subject Descriptors (according to ACM CCS): I.3.2 [Computer Graphics]: Graphics System—Standalone systems

1. Introduction

Service Level Objectives (SLOs) [AGWW06, KL03] describe the desired behavior of a service. They present agreed-upon target behaviors (e.g., capacity, performance, availability, cost) that the clients of a service can expect, and that the service provider can manage against. Both clients and service providers need a way to monitor how a service is doing against its SLOs, and display the result of that monitoring in an intuitive fashion. The focus of this paper is on a tool to perform the last of these.

SLOs can be complicated: the services we support have multiple clients, are deployed across multiple data centers (each with thousands of machines), and have multiple parts, each with multiple items, each with multiple different measurements and aspects. Each SLO line item is associated with target values and information about Service Level Indicators (SLIs). For instance, "the 50th-percentile of read operation latency should be less than 50ms" has target value 50ms and the SLI for this item is measured as a percentile value of the read operation latency distribution.

Our administrators have built a great many ad hoc dashboards, from which it is hard to build up a clear, complete, and compelling picture of what is going on quickly. Moreover, making a change to an SLO entails touching many parts of the monitoring infrastructure, and adjusting many dashboards, typically by hand. The result is systems that are needlessly hard to understand, and where the effort required to make adjustments preclude many desirable changes.

This paper describes a tool that provides SLO dashboards for services; makes it easy to navigate around the multi-dimensional space of services, their instances, users, and metrics; and is driven by the machine-readable description of the SLO itself, so that changes can be propagated automatically. The contributions of the paper include a description of the requirements and implementation of our system; the techniques used to display aspects of — and navigate around — the multi-dimensional space; and how the SLOs are used to automate the display of service-specific information.

The main idea of our system is that we provide multiple visualization components as shown in Figure 1, each of which represents specific aspects of the high-dimensional

† e-mail: hskim@ucsd.edu
‡ e-mail: sunitav@google.com
§ e-mail: johnwilkes@google.com

Figure 1: *Screenshot of the SLO visualization system. This screenshot shows where each view is placed. The topmost panel (Component 1) is a heat map showing overall SLO quality as a time series. Selecting each row updates the four panels (Component 2–5) below for the selected user. The SLO Quality table (Component 2) shows numerical values for both overall or minimum SLO quality values. The parallel coordinate (Component 3) shows SLO quality values for all SLO line items. The heat map and the parallel coordinate are designed so that they can be used two ways: displaying SLO quality values and providing a way for drill down. The bottommost two panels (Component 4 and 5) display the details as a result of visual exploration. Figure 4 shows the detail about how events update other views. Note that we erased some of our private information from the screenshot, e.g., service names, data center names, and the number of data centers.*

service outcome data. By exploring all the components together, both clients and service providers can understand what is going on in the service. In addition, each component is used as a navigational tool for narrowing down, not merely displaying information. Selections in one component update other components in a coordinated way so that all the components are simultaneously helping users understand the entire data. Note also that all the visualization views in Figure 1 are automatically generated from the SLO specification for this service.

The remainder of this paper is organized as follows: Section 2 discusses related work; Section 3 introduces the concept of Service Level Objectives; Section 4 defines the characteristics of data that we visualize in our system; Section 5 presents the visualization components that we used to solve our problem; and Section 7 discusses the implementation issues. Finally, we describe future direction of this research in Section 8 and conclude this paper in Section 9.

2. Related Work

In this section, we review previous work in visualization techniques for multi-dimensional data, as well as proposed visual analytics systems.

Parallel coordinate is a well-studied visualization technique [Ins09, Art04] and there have been many approaches utilizing parallel coordinate for visual analytics [DHNB09, RWK*06]. In SLO frameworks, clients and service providers are concerned mostly with whether the service is meeting the target. In order to visually display the status, we normalize SLIs with respect to the target values. Although SpRay [DHNB09] computed additional values from statistical analysis and visualized together for comparison, it is difficult to apply the idea directly to our system. Heat map is also a well-known technique for visualizing multi-dimensional data and for finding patterns in it. The technique showed many promising results in analyzing microarray gene expression [ESBB98, SS02]. We interpreted this heat map technology as a collection of time series and a way of navigation so that users can compare the multiple rows, each of which is a time series of SLO quality, and by choosing one row, other visualization panels are interactively updated.

There have been many applications proposing interactive visual analytics system for specific data domains [SGLS07, CGK*07, BZL*08]. Jigsaw [SGLS07] is close to our work in that the visualization starts from a stack of documents and that the system also provides multiple coordinated views. Exploring the correlation between entities is the main goal of this system and the list, graph, and scatter plot view are designed to find such relationships. However, SLO has multi-dimensional categorical data and the key problem is how to navigate to a specific part among many possible choices. The navigation decision is mainly made by quantitative values, SLO quality, not correlation between different entities.

WireVis [CGK*07] presented a visual analytics tool for identifying specific keywords within financial wire transactions. The design of WireVis is very similar to ours in that it also provided coordinate views showing different aspects of the original data and that the goal was to explore categorical and time varying data. Despite the similarity in design, the visualization techniques used in WireVis are not applicable to our problem because our data is a set of time series defined across multiple categorical data.

Barlowe el al. [BZL*08] proposed a multivariate visualization framework for high dimensional data. It focuses on partial derivatives of the dependent variables. As the three dimensions of our data, users, SLO line items and data centers, can be interpreted as categorical data, partial derivatives on our data has less meaning. This characteristic of our data makes many dimensionality reduction algorithms, such as Self Organizing Maps [KSH01], Priciple Component Analysis [M.E03], or Locally Linear Embedding [RS00], less attractive to our problems. Likewise many visualization techniques for analyzing multivariate data, for instance scatter plot matrix [CM88] or K mean clustering [Mac67], are not suitable for our problem.

3. Service Level Objectives

A Service Level Objective document represents a set of SLOs that a client and a service provider agree upon. In an SLO, both parties describe how the service will be served in terms of service performance. If a client is concerned about latency of its operations, the client may want to write down in its SLO that, for instance, 90% of requests need to be finished in 5 milliseconds. The service outcome or behavior is measured to generate a Service Level Indicator (SLI) and the value that this a service aims to deliver is called *target*. Then the client expects the service to be served as stated in the SLO and the service provider runs the service against the objectives.

Service Level Objectives Example

```
SLO SAMPLE = {
  slo_part low_latency_operation = {
    description = 'Low latency SLOs'

    // Service performance
    slo_line_item read_latency = {
      sli_ref s = {
        name = 'read_latency_global'
      }
      relation r = { comparison = 'LE' }
      target t = {
        distribution_value dv = {
          percentile p50 = { x=50, y=10e-3 }
          percentile p95 = { x=95, y=20e-3 }
          units u = {
            units = 's',
          }
        }
      }
    }
    constrainee = 'SERVICE_PROVIDER'
  }
}
```

Figure 2: *An example of Service Level Objectives that describes service level objectives for the latency of read operations. Two targets are specified: 50% of operations should be done in 10ms and 95% of operations should be finished in 20ms. The comparisons are done on latency distributions collected within a predefined window. The details of data collection are described in the SLI referenced by its name, "read_latency_global".*

An example of a trivial SLO is shown in Figure 2. The SLO example states that both parties expect 50% of read requests to be executed under 10 milliseconds, and 95% of ewS requests to be under 20 milliseconds. This example shows only a small part of the original SLO, which is able to describe various aspects of system performance, including write latency, read/write bandwidth, etc. A set of target aspects (the 50th-percentile and 95th-percentile in Figure 2) can be grouped together, called an SLO line item (read latency), and a set of SLO line items listed under the same parent, called an SLO part (low latency operation). An SLO is comprised one or more parts. Thus, SLOs have a tree structure.

Once an SLO is prepared, an SLO framework, which is part of a service, reads in and parses the SLO and starts serving the service based on the requirements described in the SLO by appropriately allocating resources. How the SLO framework controls the service is an ongoing research topic and it is beyond the scope of this paper. The SLO is machine-readable so that the job of clients is solely to write a good or correct SLO and the rest is taken care of by the SLO framework. This visualization system is also automatically created based on SLOs as a part of the framework.

The visualization system is a feedback tool that displays

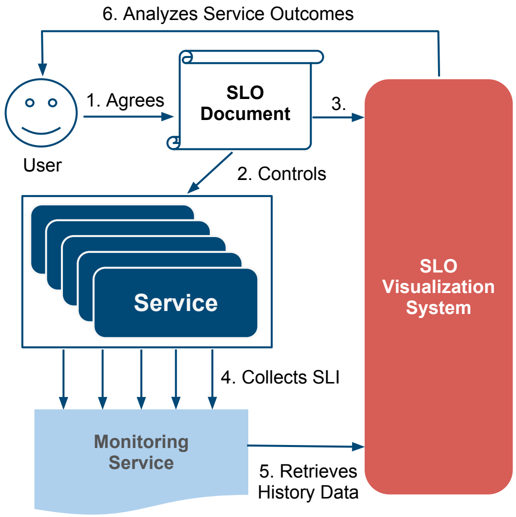

Figure 3: *Service Level Objectives Framework. First, clients agrees on their SLO documents (Step 1). The service is controlled by the SLO (Step 2) and the SLO visualization system configures visualization components based on what is described in the SLOs (Step 3). As clients use the service, historical data is accumulated in the monitoring service (Step 4). The SLO visualization system visualizes the historical data retrived from the monitoring service (Step 5). Finally, clients analyzes service outcomes (Step 6).*

the service outcome and allows clients and service providers to visually pinpoint the problematic parts of the service. In order to achieve the goal, the visualization system has to access the monitoring service collecting the service outcomes, that is, SLIs. The visualization displays the SLIs measured in many data centers across multiple continents. Because there are many aspects of service outcomes to consider, without a well-designed system, it is close to impossible for both parties to understand what is going on in the service.

Figure 3 shows the basic flow of the SLO framework. The visualization system is a part of the whole framework and interacts with many components in the framework. However, a significant portion of the framework is automated and thus, from the clients' perspective, there is minimal amount of work required. The basic use case of the SLO framework is as follows:

1. A client and a service provider agree on an SLO.
2. The service provider tunes the service resources according to the SLO.
3. The SLO visualization system reads in the SLO to create visualization components.
4. Service instance reads the SLO and produces corresponding measurements and a monitoring service collects the service outcomes from the data centers geographically distributed.
5. The visualization system retrieves the historical data from the monitoring service and visualizes the data with the information described in the SLO.
6. Both parties take actions to improve the quality of service based on the feedback from the visual exploration and revisit the SLO to get better service outcomes.

Data Axis	Notation	Description
SLO Line Item	$\mathbb{S}$	Items defined in SLO
Data Center	$\mathbb{C}$	Data centers in which this service is deployed
Time	$\mathbb{T}$	Measured every 15 seconds
User	$\mathbb{U}$	Subscribers to this service

Table 1: *Data volume. The entire data set is 4 dimensional data, SLO line item, data center, user, and time. Each SLI is measured over time at all available data centers. A service is usually described by many SLO line items and there are many users that subscribe to the service.*

4. Characteristics of Data

This section presents some important characteristics of the data that we visualize in our system. Because the data is multi-dimensional, our visualization components are designed in a way so as to help users understand the original data by looking at parts of the data.

The entire set of data we visualize, $\mathbb{M}$, is the cartesian product of multiple sets: $\mathbb{M} \triangleq \mathbb{S} \times \mathbb{C} \times \mathbb{T} \times \mathbb{U}$, where $\mathbb{S}$ denotes the set of all possible SLO line items, $\mathbb{C}$ the set of all available data centers, $\mathbb{T}$ the time intervals at which the SLI is measured, and $\mathbb{U}$ denotes the set of users. That is, a point $x = (s, c, t, u) \in \mathbb{M}$ represents SLI s in data center c at time t,of which service is for user u.

Table 1 summarizes the domain of the data measured for our visualization. The first axis of the data is SLO line items. Although the SLO line items are organized as a tree in SLOs, we simply extract the leaf nodes, each of which contains an SLI and a target value. Each SLO line item configures the monitoring service to read its SLI. The number of SLO line items in an SLO varies much but we assume it is around 10 items based on our experience in describing SLOs. The second dimension is the number of data centers. A service is usually provided at multiple geographic locations through data centers in that area. Time dimension is the third axis for our data. The monitoring service measures the SLIs periodically, e.g. every 15 seconds, from each data center. The monitoring service saves old data in secondary storage for later use as well as serving real-time data. This axis can range from an hour to a year or more. The last axis is the number of users subscribing to the services. This axis also varies service to service but we expect this number can grow to 1000 users.

Therefore, the data is multi-dimensional and the sheer volume makes it hard to understand. Administrators have designed and constructed ad-hoc dashboards to understand this

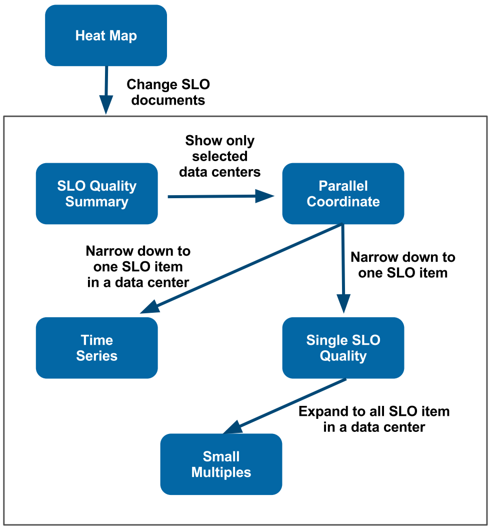

Figure 4: *Event flow between visualization components. This diagram shows how the components shown in Figure 1 interact with each other. Each component is used as a navigation tool, as well as displaying a subset of the original data.*

data, most of which show a set of time series graphs. This is certainly helpful but is not enough for clients and service providers to understand the big picture of the service because the cardinality of each axis is too big. Moreover, without an efficient way of navigating or aggregating the huge volume, it is hard to capture anomalies or trends lurking in the data.

A goal of our work is to visualize this multi-dimensional data in a clear, complete, and compelling way so that clients and service providers can quickly understand the original data and make a decision based on the visualization to improve the quality of service.

5. Visualization Components

5.1. Overview

In this section, we describe visualization components employed in our visualization system. Figure 1 shows a screenshot of our system, display five components: heat map (Secion 5.6, Component 1 in Figure 1), SLO quality summary table (Section 5.3, Component 2), parallel coordinate (Section 5.2, Component 3), time series graph (Component 5), and single SLO quality panel (Section 5.4, Component 4) in top-to-bottom, left-to-right order. An additional panel visualizes SLO data with small multiples (Section 5.5).

Each component shows a part of the original multi-dimensional data but all the components are tightly connected with each other, helping users understand what is going on in the service. An action on one of the panels updates other panels accordingly, which allows users to navigate the entire data effectively. Figure 4 shows how each component is updated when an event from another component is fired. First of all, the heat map component fires an event to make all the other parts be updated for the selected SLO. This event allows users to easily switch among multiple SLOs. Within the panels that visualize information about one specific SLO, i.e., components in the rectangle in Figure 4, the selection event fired from the SLO quality summary panel updates the parallel coordinate panel to show only a subset of data centers. By narrowing down to a subset of data centers, users select a specific data point in the parallel coordinate. The data point is described by an SLO line item and a data center. The SLO line item selected updates the single SLO quality panel to show a list of SLO quality values of that particular SLO line item across all available data centers. In addition, the selection on the parallel coordinate updates the time series chart shown in the bottom left corner in Figure 1 so that users can see the time series of their selection. From the single SLO quality panel, users can navigate to the small multiples panel to investigate the historical status of a particular data center.

All in all, the visualization components are tied together to show different aspects of the entire data set and they are constantly updated during the course of exploration and navigation. The following sections describe which part of data is displayed in each panel in details.

5.2. Parallel Coordinate Panel

Parallel coordinate [Weg90] is a well-known way of visualizing multi-dimensional data. In a parallel coordinate chart, there are many parallel axes, each of which represents one dimension of data. Each multi-dimensional data point, x, is displayed as a polyline connecting vertices on all axes. The vertex on the axis i corresponds to the coordinate of x in the i-th dimension.

Figure 5 shows an example of parallel coordinate in our visualization dashboard. Each element in the X axis represents a single aspect of an SLO line item defined in an SLO. There are 8 parallel axes in this example, thus 8 dimensional vectors are drawn in the chart. Each point on one axis, *SLO quality*, represents a normalized SLI value with respect to the target value for the SLO line item. That is, 1.0 means the SLO is exactly meeting the target, above 1.0 is exceeding the target, and below 1.0 is not meeting the target. One polyline represents SLIs measured at one data center.

Figure 5 has multiple data centers at which this service is deployed. Parallel coordinate is mainly used to find a visual pattern among multiple data points. In Figure 5, the 5th axis, "read latency 10.0ms" shows a large variance whereas points on the first and second axis show small variances. Rather than examining the variance of each item, this chart lets users find the large variance visually. In addition, this chart is also good for finding an outlier. If one line is either below or above all the other lines, one can easily find

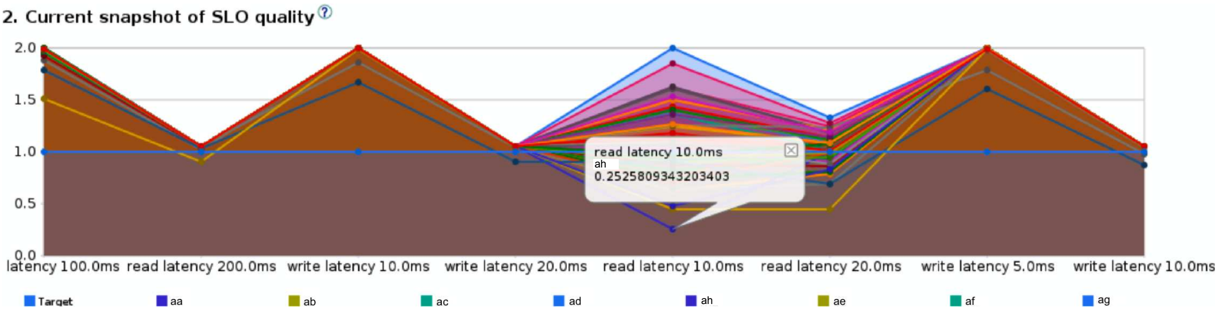

Figure 5: *Parallel coordinate for SLO line items and data centers. Each parallel axis represents one SLO line item defined in an SLO. Each instance denotes SLO quality values measured in one data center that this service is deployed. The values plotted in this chart are SLO quality values, normalized values against target values. That is, above 1.0 means the service is exceeding the expectation.*

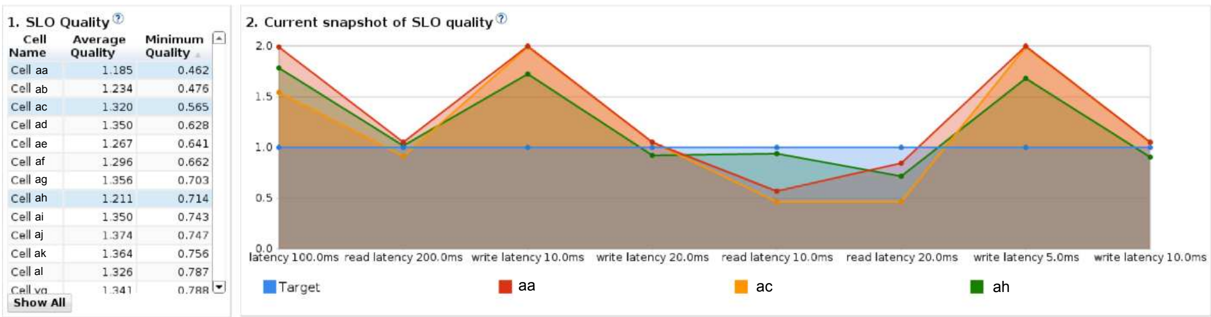

Figure 6: *SLO quality summary panel and parallel coordinate. The first, third, and eighth rows are selected in the SLO quality summary panel and the parallel coordinate panel is displaying only the data for those three selected data centers. Because the summary panel allows to sort by column, it is easy to find data centers that are not meeting the target in average or in minumum. This navigational functionality enables users to drill down to a specific data center or a specific SLO line item.*

the line. The yellow line crossing at the lowest values on the fifth and the sixth axes, for instance, shows the data center has the lowest qualities in those items as well as the first and second axes. In our system, this widget is used as a starting point for navigation. Starting from this chart, one can narrow down to a specific data center or to a specific SLO line item by selecting a point on the chart. A point on the chart is described with a data center and an SLO line item. The time series component and single SLO quality panel are updated according to the selection of a point.

5.3. SLO Quality Summary Panel

In this panel, we display a table that summarizes overall SLO quality values across all available data centers. (See Figure 6) The summary table has two columns, average quality and minimum quality, for each data center. With all SLO line items defined in an SLO, average quality values are computed as an arithmetic mean of quality values. In Figure 6, for instance, there are 8 SLO line items are shown and the values in the “Average Quality” column are the average of those 8 points. Minimum quality values show the minimum quality value among the points on the polyline. If the minimum value is above 1.0, then we know that all the SLO line items in the data center are meeting the target. One can sort this table by each column, allowing users to find the worst performing data centers in terms of average or minimum quality.

Along with the sorting, another important functionality of this table is that it allows users to select data centers of interest. In Figure 6, three data centers are selected in the SLO quality summary table. Only the corresponding lines are drawn in the parallel coordinate, data center *aa*, *ac*, and *ah*. By providing this functionality, one can narrow down to a subset of data centers in the parallel coordinate chart. A possible use case of this is that one sorts the table by minimum quality, selects data centers that have the minimum quality below 1.0, and sees how SLOs are doing in those selected data centers. Therefore, combined with the parallel coordinate chart, this table is used as a navigational widget. The quality values provide a unified measurement of comparing

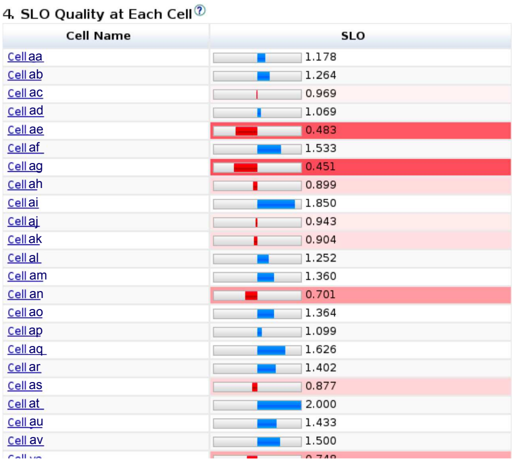

Figure 7: *Quality values for an SLO line item. This table shows quality values for a specific SLO line item across all available data centers. The table is dynamically updated when a user selects a point in the parallel coordinate. The link on the first column leads to a small multiples panel. The two visual cues, bar graph and background color help users quickly finding out which data center is not meeting the target.*

how well SLOs are doing at each data center and with the measurement, one can pick data centers of interest.

5.4. Single SLO Quality Panel

Suppose we have successfully found a data center and an SLO line item that we are interested in from the parallel coordinate, for instance, a data center that achieves lower than 1.0 quality for a specific SLO line item. There are two ways to look further into this service outcome. One is to investigate the SLO line item that caused the data center to underachieve its target and the other is to check how data center is doing as a whole. The SLO Quality Table is for the former case and the latter case is visualized by clicking on a polyline in the parallel coordinate.

Figure 7 shows a list of quality values of one specific SLO line item in each data center. This table is different from the one discussed in Section 5.3 in that this table aims to display quality values for one SLO line item, whereas the summary table shows aggregated values from multiple SLO line items. The SLO line item displayed here is chosen by a user with the parallel coordinate chart. There are two more visual cues to emphasize which data center is not meeting the target. The first visual cue is the background color in the SLO quality column. The color is assigned in proportion to the offset from 1.0. If a quality value is 0.0, then the RGB color for that data center is set to (255, 0, 0), i.e., red color, and the intensity gradually decreases between 0.0 and 1.0. The second cue is the small bar chart in each row, showing the

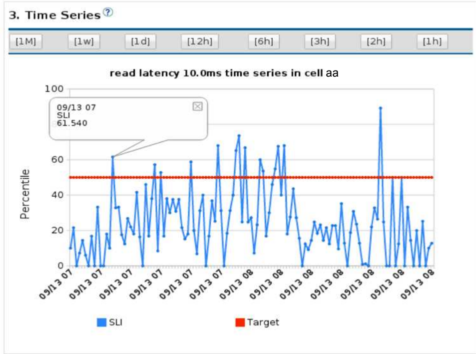

Figure 8: *Time series panel. This component plots the time series for a specific data center and a specific SLO line item. The red line in the center shows the target value and the all the points below means the service was not meeting the target.*

offset from the target. The center of the bar denotes the target value. If the service outcome is not meeting the target, the bar graph is drawn to the left in red and, if meeting, then to the right in blue. By providing these visual cues, one can easily locate the data centers that are not meeting the target.

5.5. Time Series Panel and Small Multiples Panel

The time series panel shown in Figure 8 is constantly updated based on users' selection in the parallel coordinate. While all the other components in Figure 1 displays service outcomes at the current moment, the time series actually shows historical data. The blue line shows the actual variation of service outcome over a time interval and the red line is the target. In this panel, the values are not normalized with respect to the target so that users can see the raw data.

It is often useful for both clients and service providers to find correlation between different SLO line items in one data center. For such cases, Small Multiples panel [Tuf90] displays multiple time series for SLO line items defined in the SLO to give users an idea about what is going on in that specific data center. Figure 9 shows a screenshot of a small multiples panel. This small multiples is different from the traditional one because it also shows the tree structure embedded in an SLO. As a result, each SLO document will produces a different small multiple view. Each time series graph is located at the leaf node of the SLO tree structure. In this example, there are two SLO parts, "Normal Case" and "Low Latency", and both have read and write latency. Each latency SLO line item has two targets, producing two time series. Because users are often interested in a subset of graphs, the tree can be expanded or collapsed. By hiding less important graphs, one can compare only the interesting SLO line items more closely.

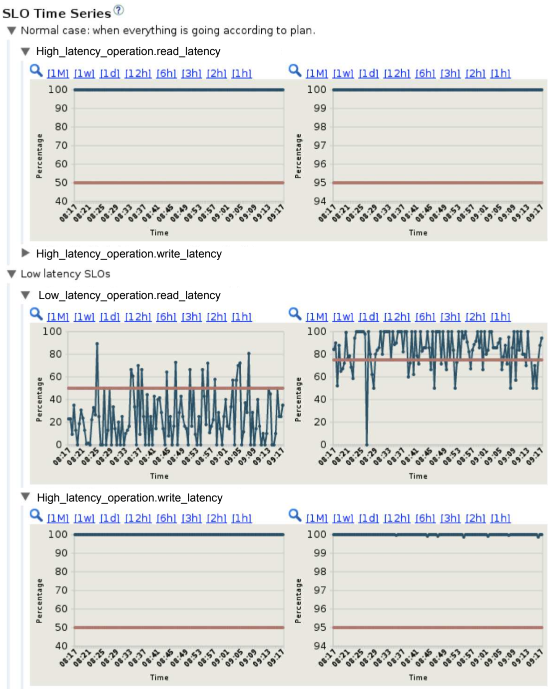

Figure 9: *Small multiples combined with the tree structure in SLO. In this example, There are two SLO part nodes under the root node. Each SLO part has read and write latency SLO line item. Each SLO line item is evaluated with two target values. The tree can be expanded or collapsed for comparison.*

5.6. Heat Map Panel

We have discussed visualization components to find an interesting SLO line item or an underperforming data center and the discussion so far assumes that we have only one SLO. However, in reality, there are many clients subscribing to the service. Therefore, it is important for service providers to have a way to explore how all the users are doing. As discussed in Section 4, the data for one user is represented by 3D data and we need a way to aggregate this information into a data set representing the status of service outcomes for multiple users. For this goal, we employ a heat map [ESBB98]. The cell at coordinate (t,u) represents the overall SLO quality for user u at time t. The overall SLO quality is defined as in the SLO summary table discussed in Section 5.3. An example of heat map is shown in Figure 1. The green color indicates that the overall SLO quality is above 1.0, i.e., the service outcomes are exceeding the targets in average, the black color means that the quality is 1.0, and the red color means it is below 1.0. The color at each cell is interpolated between the three colors depending on the actual SLO quality value.

The heat map visualizes a correlation between multiple users. If an SLO is not meeting the target in terms of overall quality, it is likely to affect other users, too. Service providers are most interested in these types of events and heat maps give a way to compare across multiple users. Another important feature of this component is that this visualization is also used as a navigation tool. The system dynamically updates the bottom part of Figure 1. When a row in the heat map is selected, the bottom part changes to display information about the selected SLO.

	SLO Item	Data Center	Time	Component
1	s	c	t	N/A, Trivial
2	s	$\mathbb{C}$	t	Single SLO Quality (4)
3	$\mathbb{S}$	c	t	N/A
4	$\mathbb{S}$	$\mathbb{C}$	t	Parallel Coordinate (3)
5	$\sum s$	$\mathbb{C}$	t	SLO Quality Summary (2)
6	s	c	$\mathbb{T}$	Time Series (5)
7	$\mathbb{S}$	c	$\mathbb{T}$	Small Multiples
8	s	$\mathbb{C}$	$\mathbb{T}$	N/A
9	$\mathbb{S}$	$\mathbb{C}$	$\mathbb{T}$	N/A, entire data

Table 2: *Data slicing: each component visualizes a part of the entire data set. $\mathbb{S}$ denotes that the entire elements in the SLO line item are displayed. $\mathbb{C}$ and $\mathbb{T}$ is also defined likewise. t is set to the current time. If a particular element is picked either in the SLO line item domain or in the data center domain, s or c is used to represent such cases, respectively. The number in the last column shows the component number in Figure 1.*

6. Analysis

All the visualization components discussed so far are designed in such a way that users can easily understand the 4D data set $\mathbb{M}$. We focus on a 3D data set, $\mathbb{M}_u = u \times \mathbb{S} \times \mathbb{C} \times \mathbb{T}$, by narrowing down to a specific user, i.e., a specific SLO. The heat map component provides an effective way of investigating across multiple users and allows users to switch between many SLOs. The remaining question is then how to effectively visualize $\mathbb{M}_u$.

Our approach is that we slice the data and visualize an aggregated subset of the original domain. Table 2 lists which portion of data is visualized by each visual component. Obviously a single point (Row 1) or the entire data set (Row 9) is usually not interesting. The time axis is fixed either as t, usually set to the current time, or $\mathbb{T}$ because the current status is the most important information to service providers. If one is interested in past events, we rather provide as a time series instead of point-wise navigation.

For SLO line items and data centers, both are important to clients and service providers and it is essential to narrow down to a specific SLO line item or a particular data center. This navigation allows user to find SLO line items that are not meeting the targets or data centers that are not performing well. Therefore, providing views for 1) a single SLO line item in all data centers (Row 2, Component 4 in Figure 1) and 2) a time series of an SLO line item in a specific data center (Row 6, Component 5) is equally important. In order to find correlation between multiple SLO line items, small multiples visualizes all SLO line items in a specific data center (Row 7). The single SLO quality panel is specifically designed for the first view but the parallel coordinate (Row 4, Component 3) and SLO quality summary panel (Row 5, Component 2) provide a big picture for comparison between SLO line items. The data that the single SLO quality panel visualizes is a subset of that of both parallel coordinate panel and SLO quality summary panel. The small multiples panel handles the second case, displaying information of all SLO line items in a specific data center.

We do not provide a view for $\mathbb{D} = \{\mathbb{S} \times c \times (t = 0)\}$ (Row 3) because the parallel coordinate panel (Row 4) already visualize a super set of this case. For the view for $\{s \times \mathbb{C} \times \mathbb{T}\}$ (Row 8), we could visualize with multiple instances, each of which represents a time series measured in a specific data center, in the time series panel. However, we decided not to include this in our visualization because it does not add much insight to users.

7. Implementation

We built our proposed system as a web application so that any users in our environment can easily access with a web browser. The standard operations work as HTTP request and response communication. However, due to the dynamic updates among visualization components, asynchronous data pulling and update were heavily used with the help of Asynchronous Javascript And XML (AJAX) technology [Pau05].

We optimized the system to minimize the round trip communication when an update is needed. For example, suppose a user selects a point in the parallel coordinate panel, which fires an event to make the single SLO quality panel update its information. In this case, because all the necessary data is received in the browser, we simply update the table using the cached data. Likewise, when a row is selected in the SLO quality summary table, there is no communication between the browser and the server to update the parallel coordinate panel. The panel simply re-draws the panel using the data already received from the server when the page is loaded for the first time. Another layer of caching is done at the server. Because the same data is likely to be requested again in a short period time, the server stores the data that it retrieved from the monitoring service for 30 seconds. The server cache saves the communication between the server and the monitoring service and re-executes the same query at the monitoring service.

One characteristic of our system is that we employed several technologies from open source projects: Google Web Toolkit (GWT)Google Visualization (GViz), and Protocol Buffer.

GWT [HT07,Goob] is a development software framework that allows developers to write AJAX software in Java language. The compiler in GWT converts the Java code into Javascript code, which overcomes many difficulties often arising during Javascript developments. In particular, the compiler takes care of the compatibility issue across many existing browsers. GViz [Gooa] provides a set of visualization toolkits which we aggressively used for our visualization components. GViz is supported in GWT via Javascript Native Interface (JSNI), which allows Java developers to embed Javascript native code in a Java program. Protocol Buffer [Gooc], also known as *protobuf*, is a framework for data exchange and we used this technology to parse SLOs. We defined SLO documents as protobuf messages. Our visualization system reads in the file and extracts information about SLO line items, as well as in which server the data stream is recorded and served. The contents of the visualization dynamically change depending on which SLO it reads. All those open source projects enabled our visualization system to be easily served through web browsers.

8. Future Work

The current SLO visualization system is designed to navigate multi-dimensional historical data. As a next step, we are looking into data about the future. One of the features that we would like to see in our system is a what-if analysis. Service providers want to see what would have happened if the target values had been set differently. Even clients would be interested to see if their services could have obtained better performance with different target values or with a completely different set of SLO line items. Visualizing the trend of SLI data, whether increasing or decreasing, will also be an important part in the what-if analysis.

Another direction we plan to investigate is interactively, graphically modifying SLOs. The current system reads SLOs and automatically generates all the visualization components but there is no way of changing the original SLOs. Either a client or a service provider may realize that the target values does not reflect their expectation after exploring the history data from our visualization system. In that case, one may want to change the values to a better value interactively. Adding a new set of SLO line items is also an important part for modifying SLOs.

9. Conclusion

We have presented a visual analytic system that visualizes Service Level Objectives. Service outcomes are measured and collected by monitoring service and our goal was to provide a visual analytics tool that allows clients and service providers to easily explore and navigate the service outcomes data. Due to the high dimensionality of the data, we provided several information visualization components, including parallel coordinate, table, small multiples, and heat map. All the components are designed in a way that users can quickly narrow down to interesting parts of the service as well as capture the current status of the service.

References

[AGWW06] AUYOUNG A., GRIT L., WIENER J., WILKES J.: Service contracts and aggregate utility functions. In *Proceedings of the IEEE Symposium on High Performance Distributed Computing* (2006), pp. 119–131. 1

[Art04] ARTERO A. O.: Uncovering clusters in crowded parallel coordinates visualizations. In *Proceedings of the IEEE Symposium on Information Visualization 2004* (2004), IEEE Computer Society, pp. 81–88. 2

[BZL*08] BARLOWE S., ZHANG T., LIU Y., YANG J., JACOBS D.: Multivariate visual explanation for high dimensional datasets. In *Visual Analytics Science and Technology, 2008. VAST '08. IEEE Symposium on* (October 2008), pp. 147–154. 3

[CGK*07] CHANG R., GHONIEM M., KOSARA R., RIBARSKY W., YANG J., SUMA E., ZIEMKIEWICZ C., KERN D., SUDJIANTO A.: WireVis: Visualization of categorical, time-varying data from financial transactions. In *Visual Analytics Science and Technology, 2007. VAST 2007. IEEE Symposium on* (October 2007), pp. 155–162. 3

[CM88] CLEVELAND W. C., MCGILL M. E.: *Dynamic Graphics for Statistics*. CRC Press, Inc., Boca Raton, FL, USA, 1988. 3

[DHNB09] DIETZSCH J., HEINRICH J., NIESELT K., BARTZ D.: SpRay: A visual analytics approach for gene expression data. In *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on* (October 2009), pp. 179–186. 2

[ESBB98] EISEN M. B., SPELLMAN P. T., BROWN P. O., BOTSTEIN D.: Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci U S A 95*, 25 (December 1998), 14863–14868. 3, 8

[Gooa] GOOGLE INC.: Google Visualization API. `http://code.google.com/apis/charttools`. 9

[Goob] GOOGLE INC.: Google Web Toolkit. `http://code.google.com/webtoolkit/`. 9

[Gooc] GOOGLE INC.: Protocol Buffers. `http://code.google.com/p/protobuf/`. 9

[HT07] HANSON R., TACY A.: *GWT in Action: Easy Ajax with the Google Web Toolkit*. Manning Publications Co., Greenwich, CT, USA, 2007. 9

[Ins09] INSELBERG A.: *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2009. 2

[KL03] KELLER A., LUDWIG H.: The WSLA framework: Specifying and monitoring service level agreements for web services. *Journal of Network and Systems Management 11* (2003), 57–81. 1

[KSH01] KOHONEN T., SCHROEDER M. R., HUANG T. S. (Eds.): *Self-Organizing Maps*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001. 3

[Mac67] MACQUEEN J.: Some methods for classification and analysis of multivariate observations. In *Proc. of the Fifth Berkeley Symposium on Mathematical Statistics and Probability* (1967), pp. 281–297. 3

[M.E03] M.E. T.: Principal component analysis (2nd ed.). I. T. Jolliffe. *Journal of the American Statistical Association 98* (January 2003), 1082–1083. 3

[Pau05] PAULSON L.: Building rich web applications with ajax. *Computer 38*, 10 (October 2005), 14–17. 9

[RS00] ROWEIS S. T., SAUL L. K.: Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science 290*, 5500 (2000), 2323–2326. 3

[RWK*06] RÃIJBEL O., WEBER G. H., KERÃĎNEN S. V. E., FOWLKES C. C., HENDRIKS C. L. L., SIMIRENKO L., SHAH N. Y., EISEN M. B., BIGGIN M. D., HAGEN H., SUDAR D., MALIK J., KNOWLES D. W., HAMANN B.: Pointcloudxplore: Visual analysis of 3d gene expression data using physical views and parallel coordinates. In *Eurographics/IEEE-VGTC Symposium on Visualization Proceedings* (2006), pp. 203–210. 2

[SGLS07] STASKO J., GORG C., LIU Z., SINGHAL K.: Jigsaw: Supporting investigative analysis through interactive visualization. In *Visual Analytics Science and Technology, 2007. VAST 2007. IEEE Symposium on* (October 2007), pp. 131–138. 3

[SS02] SEO J., SHNEIDERMAN B.: Interactively exploring hierarchical clustering results [gene identification]. *Computer 35* (July 2002), 80–86. 3

[Tuf90] TUFTE E. R.: *Envisioning Information*. Graphics Press, Cheshire, CT, 1990. 7

[Weg90] WEGMAN E. J.: Hyperdimensional data analysis using parallel coordinates. *Journal of the American Statistical Association 85*, 411 (September 1990), 664–675. 5