BOM-Vis: A Visualization of Network Health and Status

CPSC 547 Project Proposal

Dennis Park dennispk@cs.ubc.ca

Domain Description

BankWorld is a planet much like Earth, identical in size, but with a geography consisting of a single land mass in which a handful of nation-states exist side by side. The Bank of Money (BOM) is perhaps the most important organization within BankWorld, having its facilities spread out all across the globe. BOM is organized into a collection of regions: large regions, small regions, and HQ, each populated by various facilities such as data centres (overseen by HQ), regional headquarters, and branches of the bank. Each of these facilities in turn are populated by various types of machines, which include servers, workstations, and ATMs.

The purpose of this project is to design and implement a visualization tool, which can be used to monitor the health and status of BOM's network. In particular, the tool aims to support network administrators in identifying and diagnosing anomalies in the network's behaviour and status.

The dataset for BOM's network is provided by the IEEE Conference of Visual Analytics Science and Technology (VAST) as part of their 2012 visualization challenge. Two separate datasets are provided: (i) a catalogue of BOM's 895,025 machines, detailing their locations, IP addresses, types, and functions; (ii) and a log of status reports spanning two days, which describes machine health, activity levels, and general status across the network.

The following provides a detailed listing of the core tasks I intend to support:

- Identify regions experiencing significant levels of unscheduled downtime at a given moment in time.
- Having identified a region experiencing significant levels of unscheduled downtime, identify when it began and when it ended (i.e. identify the interval of anomaly).

- Identify regions experiencing significant levels of policy deviation at a given moment in time.
- Having identified a region experiencing significant levels of policy deviation, identify when it began and when it ended (i.e. identify the interval of anomaly).
- Having identified an interval of anomaly w/r/t policy deviation, profile its growth (e.g. where it began, how it spread from facility to facility and region to region).

- Identify regions experiencing significant levels of full cpu consumption (relative to other regions) at a given moment in time.
- Having identified a region experiencing significant levels of full cpu consumption (relative to other regions), determine whether the levels are anomalous for the given region by viewing it within the context of time (i.e. does it deviate from the region's normal levels of cpu consumption?).
- Having confirmed that a region experienced anomalous levels of full cpu consumption, identify when it began and when it ended (i.e. identify the interval of anomaly).

- Identify regions experiencing anomalous levels of machine connections (relative to other regions) at a given moment in time.
- Having identified a region experiencing anomalous levels of machine connections (relative to other regions), determine whether the levels are anomalous for the given region by viewing it within the context of time (i.e. is the timepoint of focus a spike or dip breaking the general pattern of connection levels in the given region?).
- Having confirmed that a region experienced anomalous levels of machine connections, identify when it began and when it ended (i.e. identify the interval of anomaly).

- Identify an interval of anomaly w/r/t machine connection levels, which breaks the general pattern (i.e. identify spikes and dips breaking the general pattern of connection levels). Similarly, identify an interval of anomaly w/r/t unscheduled downtime, policy deviation, and full cpu consumption levels.
- Having identified an interval of anomaly w/r/t machine connection levels, localize the anomaly to a particular region or group of regions. Similarly, having identified an interval of anomaly w/r/t/ unscheduled downtime, policy deviation, or full cpu consumption levels, localize the anomaly to a particular region or group of regions.

Personal Expertise

The only background I have of network monitoring is a course on computer networks (CPSC 317), which I took during my undergraduate program. Beyond this, I have no experience in the actual activity of network monitoring, nor have I ever designed network monitoring tools. I have limited experience using D3.js and moderate experience in general web development.

Proposed InfoVis Solution

As previously mentioned, the purpose of this project is to design and implement a visualization tool, which can be used to monitor the health and status of BOM's network. The proposed solution aims to achieve this by a combination of two main views—one providing a dynamic picture of how the network changes over time, and another providing a static picture of the network at a single moment in time—presented one below the other. These views are described next.

- **Static View.** This view provides an overview of the network's health and status at a given moment in time. It consists of a BankWorld map partitioned into its various regions populated by small points to represent facilities, along with a set of radio buttons which can be toggled to focus on different aspects of the network's status.

  When the option for *unplanned downtime* is selected, the various facilities of BOM become encoded by a linear colour map to represent the aggregate number of machines housed in each facility that are experiencing unplanned downtime.

  When the option for *policy deviation* is selected, the facilities become encoded by a combination of mark size and mark colour to represent the aggregate number of machines experiencing each level of policy deviation. More specifically, a given facility will be represented by a bullseye of potentially 4 concentric circles, the area of each ring representing the aggregate number of each 4 levels of policy deviation. The rings will be colour coded along a linear progression.

When the option for *full cpu consumption* is selected, the facilities become encoded by a linear colour map to represent the aggregate number of machines housed in each facility that are experiencing full cpu consumption.

When the option for *machine connections* is selected, the facilities become encoded by a linear colour map to represent the mean connection level of the machines housed in each facility.

Beyond the features described above, the static view will also interact with the *dynamic view* (described below) to navigate along different points in time. By selecting different columns on the *dynamic view* or by using the left and right arrow keys, the user can jump between different points in time as well as move sequentially along it (if time becomes limited, only one of these navigations will be implemented). The currently selected timepoint will be represented by a vertical line over the dynamic view.

- **Dynamic View.** This view provides a representation of the network's health and status across time. It consists of a horizontal timeline representing the 48 hours of activity logged in the provided dataset. As with the *static view* described above, the dynamic view changes in response to user toggling of the radio buttons to show different aspects of the network.

  The dynamic view is also dependent on user interaction with the *static view* in the following way: when users click on a region within the static view, it becomes the focus of the dynamic view, showing time-dependent data for just that region. Multiple regions can be selected, in which case the dynamic view shows time-dependent data for the aggregate of the regions. Initially, all regions are unselected, and the dynamic view shows time-dependent data aggregating the entire network.

  When the option for *unplanned downtime* is selected, the dynamic view shows a bar chart encoding the unplanned downtime levels for the selected region(s) as a function of time.

  When the option for *policy deviation* is selected, the dynamic view shows a stacked bar chart encoding each level of policy deviation for the selected region(s) as a function of time.

  When the option for *full cpu consumption* is selected, the dynamic view shows a bar chart encoding the *full cpu consumption* levels for the selected region(s) as a function of time.

  When the option for *machine connections* is selected, the dynamic view shows a line chart encoding the *machine connection* levels for the selected region(s) as a function of time.

Although there are a lot of implementation required for the features above, luckily the 4 options for both the static and the dynamic view (i.e. unplanned downtime, policy deviation, full cpu consumption, and machine connection) are presented in nearly identical ways so that a lot of the work can be easily duplicated. Nevertheless, I will first focus on having views for the policy deviation option as the top priority, and then once these are done, I will move onto implementing the other options.

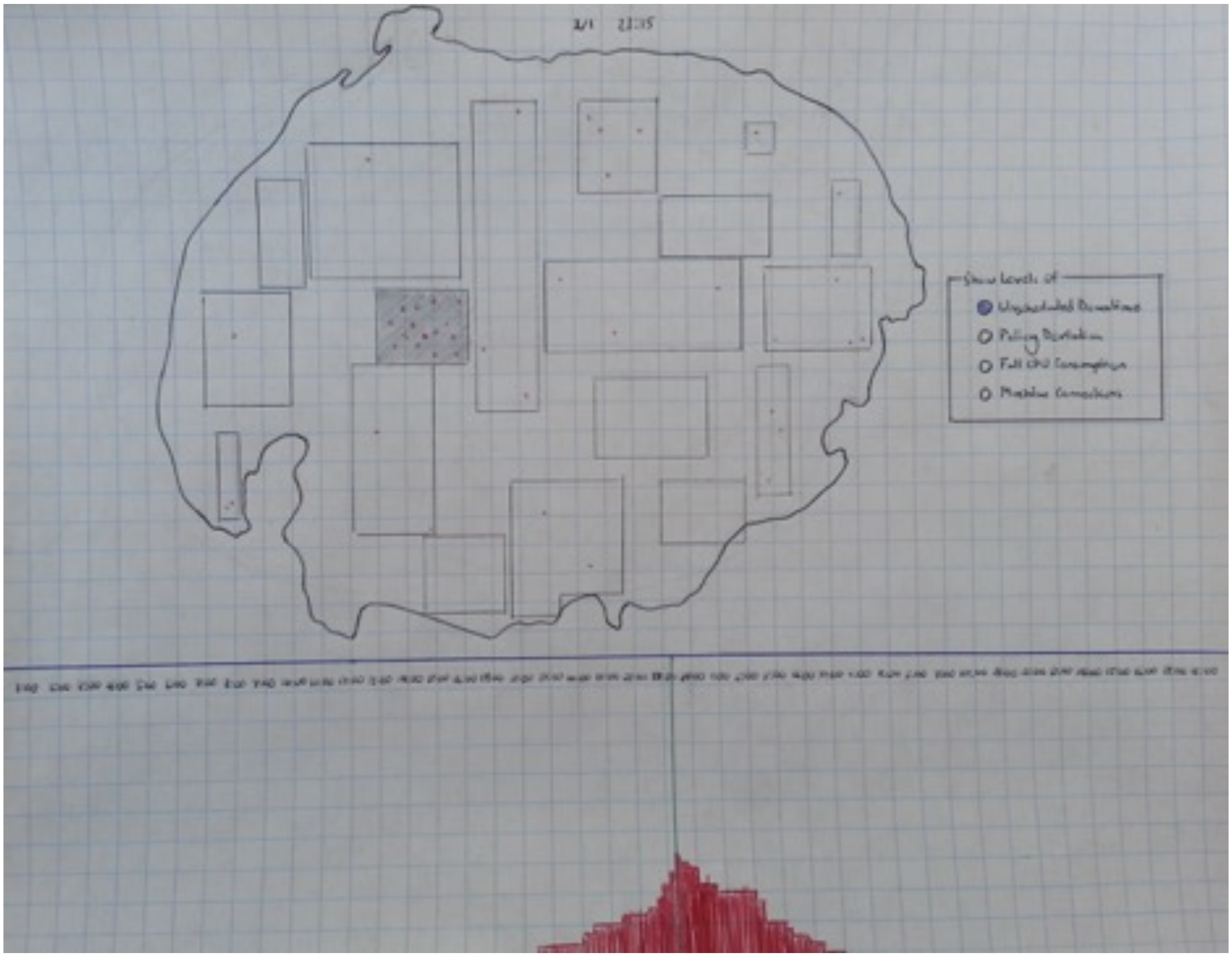

Figure 1: Interface with *Unscheduled Downtime* and a single region selected.

Scenario of Use

On February 2, 2012, the Bank of Money network experienced network wide outage. John has been brought in to find out how it may have happened.

He starts by running BOM-Vis. Once it loads, he moves to the very last recorded status report by clicking the far right end of the dynamic view. He then toggles through the various options (*unplanned downtime*, *policy deviation*, *full cpu consumption*, *machine connections*), getting an understanding of the network's state at this last timepoint. He toggles back to the *policy deviation* option. In this last timepoint, the network seems to have experienced high levels of policy deviations all across the network. By looking through the dynamic view, John tries to identify when the policy deviations began.

Policy deviations seem to first occur sometime around 1:00 pm on February 2, and climb steadily. John moves to near the beginning of the policy deviations by pressing the corresponding timepoint in the dynamic view. At this point, the static view shows only a handful of facilities experiencing policy deviations, which mainly consist of level 1 and some level 2 deviations, and these seem to be clustered around regions 3 and 4. John selects these two regions on the static view so that only their data is shown on the dynamic view. He toggles through the other options (i.e. full cpu consumption, machine connections, etc.) again, trying to find any other anomalies occurring in these two regions around this time.

John finds that around 15 minutes before the policy deviations began, there was a small spike of unplanned downtime. He moves to this timepoint in the dynamic view, and finds that these unplanned downtime were localized in a single facility within region 3. He finds the address of this facility and leaves to ask personnel there regarding the event.

Proposed Implementation Approach

I propose to build the system as a web application using the D3.js library on the front end and a basic Node.js server on the backend to talk with the MySQL database, in which I will store the network data. A map of BankWorld is provided by VAST, and I will simply use this as is for the static view.

Milestones and Schedule

November 14 Have a basic server which loads a rudimentary webpage (i.e. has structure for the two views, the controls, and loads the map into the static view) although most of the functionality will likely not be implemented.

November 21 Complete static view for the policy deviation option.

November 28 Complete dynamic view for the policy deviation option.

December 5 Complete remaining options for static view.

December 12 Complete remaining options for dynamic view.

December 15 Complete presentation slides (deadline for presentation)

December 17 Complete paper (deadline for paper)

Previous Work

Network visualization is a widely studied topic, and there are thousands of visualization tools supporting network monitoring and analysis, provided both commercially and open source [1]. These tools range both in the kind of data they visualize (many of them dealing with how traffic moves along a network) and the techniques they employ [1]. As part of a larger class of network monitoring tools, a large group of network visualization tools focus on the use case of detecting and analyzing network security threats [2].

With regard to the dataset and tasks dealt by the current project, there is also a wealth of existing systems. As part of the VAST 2012 mini-challenge, 27 different solutions are provided online [3].

References

[1] A Survey of Network Traffic Monitoring and Analysis Tools www.cse.wustl.edu/~jain/cse567-06/ftp/net…monitors3/index.html

[2] Kiran Lakkaraju, William Yurcik, and Adam J. Lee. Nvisionip: netflow visualizations of system state for security situational awareness. In Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security, VizSEC/DMSEC '04, pages 65–72, New York, NY, USA, 2004. ACM.

[3] Visual Analytics Benchmark Repository http://hcil2.cs.umd.edu/newvarepository/VAST... 2012/challenges/MC1…20Cyber/