ALAMA: App Logs as Measurement Approximation

Naveen Nandan, Wang Ting, and Daniel Dahlmeier

***Abstract*—The penetration of mobile devices in our daily lives is becoming quite significant with the advancement in technology. Data in the form of logs generated from user interaction with various apps is increasing at an alarming rate. The system developed demonstrates real-time analytics and visualization of data from mobile app logs for traffic and weather. The methodology used goes by the approximation that the app logs are an indication of the actual measurement of traffic and weather conditions, if not exact. The system architecture illustrates the capability to easily integrate actual traffic and weather data sources if made available or, in general, the capability of the system to handle any spatio-temporal data sources.**

***Index Terms*—Context-based services, real-time analytics, real-time data visualization, urban data mining.**

I. Introduction

Today, we see an exponential growth of smart phones and related apps in the market. Owners and consumers of these devices and apps are knowingly or unknowingly contributing a vast amount of data about their usage of these apps. Often, these apps directly store all this information about users, with their consent, into traditional databases or the consumer's app usage details as log files [1]. We make use of log files generated from two such apps TrafficLah and WeatherLah developed by buUuk, Singapore which contain usage details of the app mapped to the time and location, and no information of the user as an individual in any form.

Most of the consumer apps today are targeted to provide context and location-based services to their users and hence, these generate logs that contain collective spatial-temporal records [2]. This could be very useful in studying the aggregated behavior or even forecasting the behavior of a population. In an urban context, it is very interesting to study such aggregated behavior as this mostly influences the dynamics of the city such as people movement, traffic situations, weather conditions, and so on [3].

The data used for this paper, to validate the developed methodologies and system, was made available as a result of the Clean & Green Hackathon 2013, an initiative organized together by the National Environment Agency, Singapore and Newton Circus. The initial prototype was designed, implemented and presented in a period of 36 hours.

Manuscript received June 15, 2013; revised August 5, 2013. This research was supported with the funding from the Economic Development Board and the National Research Foundation of Singapore.

Naveen Nandan, Wang Ting, and Daniel Dahlmeier are with Research & Innovation, SAP Customer Engagement & Strategic Projects, Singapore (e-mail: {naveen.nandan, dean.wang, d.dahlmeier}@sap.com).

II. Traffic Incidents vs. App Logs

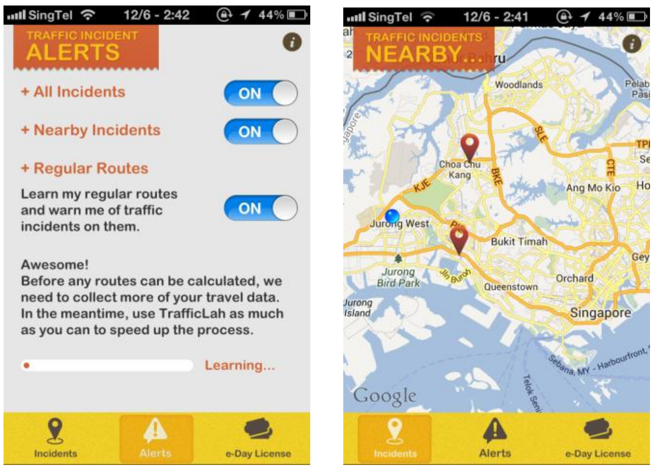
In this section we describe the iPhone app TrafficLah created by buUuK Pte Ltd and how do we use their log as an indication of the traffic condition.

A. The TrafficLah App

TrafficLah is an app that helps people to avoid getting stuck in traffic jams by reporting nearby traffic condition of the user. Fig. 1 shows a screen shot of the app.

The log file of TrafficLah app is not an accurate record of the actual traffic condition, instead, it is a log of when and where do people open the app and check the traffic condition. We argue that:

1) A user would be more likely to open the app and check for traffic condition when he is expecting some heavy traffic ahead, or he is already stuck in some traffic jam.
2) When a traffic jam is formed up, the density of drivers and cars per unite space will increase. Thus the density of the log entries in this area at this certain time would also increase.
3) If a user opens the app from an alert, it is highly likely that she/he is near a place with bad traffic condition.
4) If at certain point of time for some area there is no log of using the app exists, it either shows that there is nobody checking the traffic or there is no people at all. In either case it could be an indication of good/smooth traffic condition.



+ All Incidents ON
+ Nearby Incidents ON
+ Regular Routes
Learn my regular routes and warn me of traffic incidents on them. ON

Awesome!
Before any routes can be calculated, we need to collect more of your travel data. In the meantime, use TrafficLah as much as you can to speed up the process.

Learning...

Fig. 1. Screen shots of the TrafficLah App (iPhone version).

We could assume that the spatial-temporal density of the logs is an indication of the traffic condition. We are therefore motivated to find out the exact connection between the log and the actual traffic condition.

B. Traffic Incidents vs. App Log

We obtained actual traffic incidents statistics from Land and Transportation Authority (LTA) of Singapore. However, due to the lack of quantitative measurements of traffic condition, we choose to figuratively demonstrate the correlation between the density of App Logs and traffic incidents.

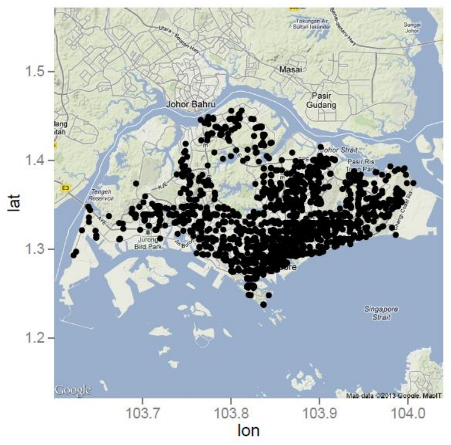


Fig. 2. Plot of log locations of TrafficLah App.

Using R, we are able to import both data sets and plot them on the map of Singapore.

We note due to the overwhelming size of the TrafficLah log, we are taking only 2000 randomly sampled log entries to plot Fig. 2.

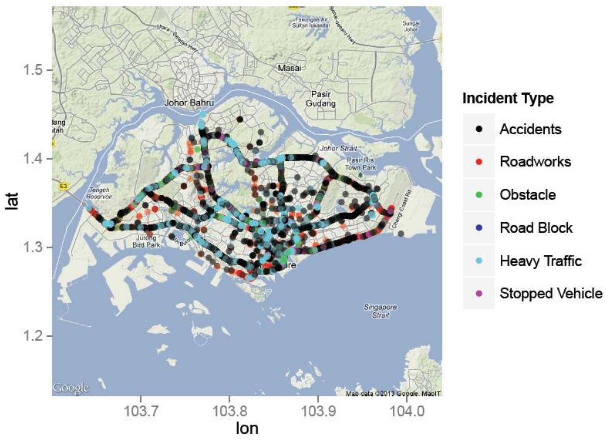


Fig. 3. Plot of incidents locations and types.

In Fig. 3, we use different colors to differentiate the types of traffic incidents. We can see most of the incidents take place in the city center, or on the high ways.

We can see that although all the accidents take place on the roads, the logs include massive entries off the roads. Therefore directly relate the locations between these two data sets may be inaccurate and less meaningful.

Moreover, based on our arguments in the previous section, the traffic condition is more related to the density of the logs instead of the locations of them. Therefore, we draw the density heat map of the same data in Fig. 4.

From Fig. 4 we can see that the density near the city center is the highest, which corresponds to the traffic incidents plot.

To have an even better view on these data, we overlay the incident plot on the heat map of the logs, and plot hourly in Fig. 5. It shows the app log density and traffic incidents in each hourly time window. We can clearly identify the morning and evening rush hour in 8 am and 6 pm respectively. We can also see that road works are only conducted in 0 am or 9 pm, when the traffic condition is generally good. More importantly, we can see that the density of the logs is correlated with the density of the incidents, which confirms with our argument and assumption earlier. Thus, using the app log, we are able to monitor the traffic condition of the city.

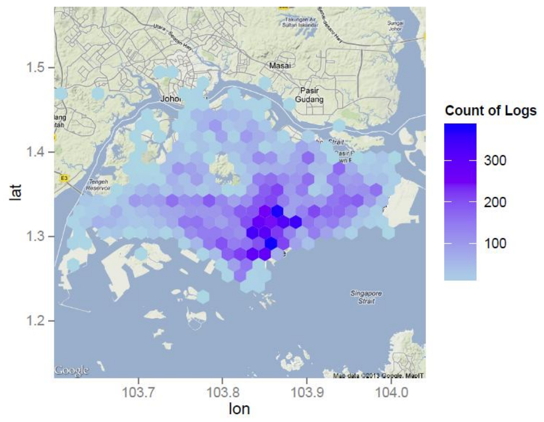


Fig. 4. Density of logs of TrafficLah App.

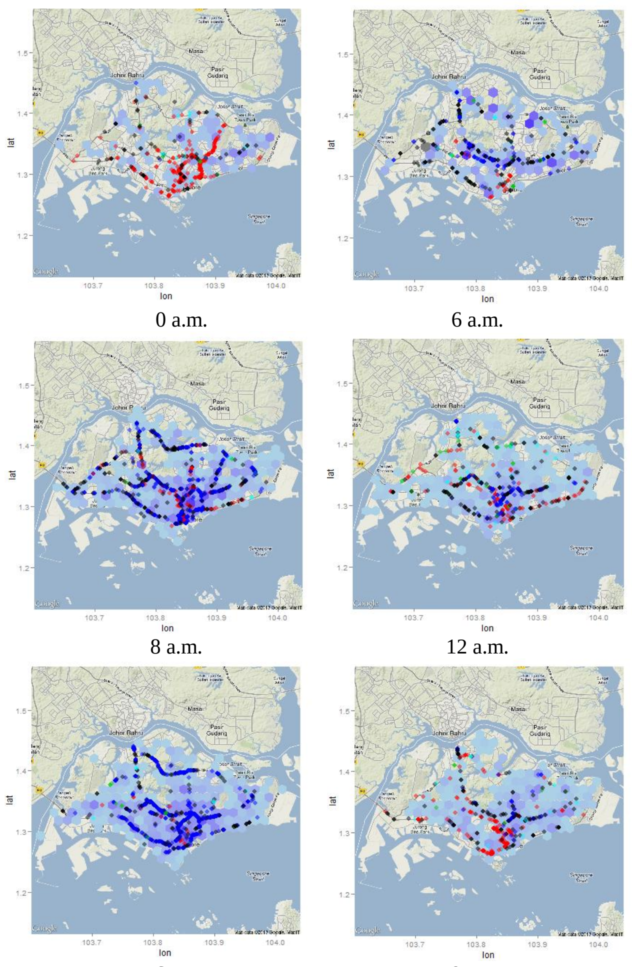


Fig. 5. Hourly plot of incidents and log heatmap.

III. Extending Our Work to Weather

One interesting extension of this piece of work is to use the same methodology to another similar app, namely WeatherLah, also created by buUuK. Fig. 6 shows some screenshots of the app.

WeatherLah uses crowd sourcing to collect data from users and update the weather condition. It keeps a similar log data set as TrafficLah, except having one extra field to store the reported weather condition.

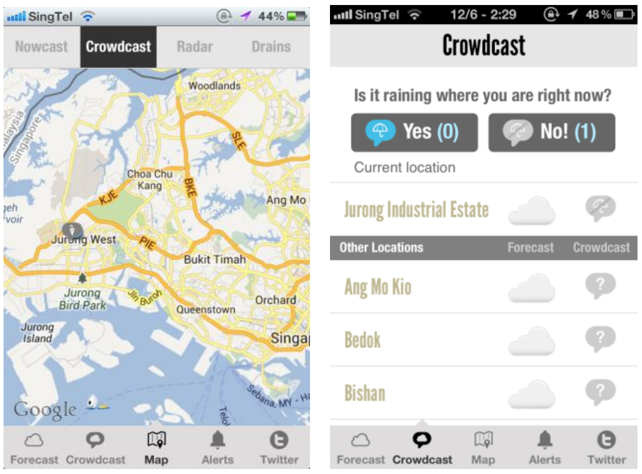


Fig. 6. Screen shots of the WeatherLah App (iPhone version).

It has been discussed widely that rain could lead to traffic jams and incidents [4]-[6]. Our next step is to study the correlation between the WeatherLah logs and TrafficLah logs. Maybe in the near future, we can forecast the traffic condition from the weather, using only logs of apps.

IV. System Architecture

The system developed consists of the following components [7]:

- *Event generator*
- *Event stream processing engine*
- *Query result subscriber*
- *Websocket bridge*
- *Visualization*

A. Event Generator

One of the important requirements to develop/validate this system was to be able to generate the events at the exact same rate as it occurred and maintain the inter-arrival time between events. In order to achieve this, a multi-threaded scheduler (as shown in Fig. 7) was implemented that could read off the file containing the app logs and asynchronously publish it to the event processing engine.

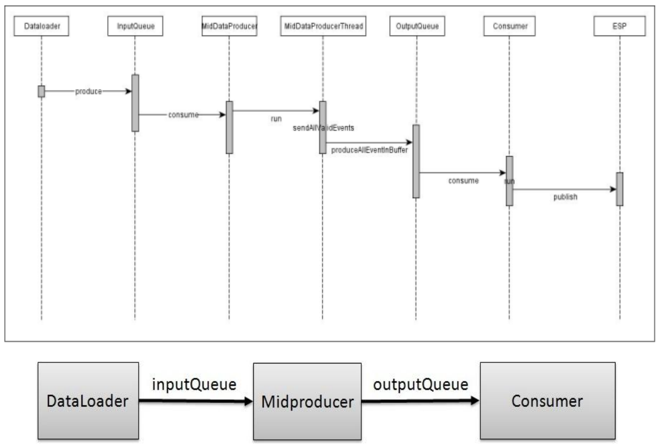


Fig. 7. Event generator design.

B. Event Stream Processing Engine

For performing stream analytics, we use the SAP Sybase Event Stream Processor. In order to achieve high performance, the queries are broken down into multiple small operators, with each flow of operators cascaded to produce a desired result. Each operator is custom built using a built-in scripting language called stream processing language shell (SPLASH) that helps to extend the engine's query language called continuous computation language (CCL).

C. Query Result Subscriber

The query result subscriber is an interface between the event processing engine and the external visualization stack. It is a simple listener API that gets notified every time there is an update in the result stream.

D. Websocket Bridge

After evaluating multiple visualization frameworks, the challenge was to ensure high performance, both in terms of rendering the visualization and making the data points available at the visualization layer. Since the intention was to design the system to handle both high volume and high velocity data, the best method was to be able to push the events into the visualization layer, instead of the visualization stack having to poll the events. For this, we make use of the Websocket implementation of the HTML5 stack, which is a high speed transfer protocol for the web.

E. Visualization

Two categories of visualizations are generated: 1) a real-time analytical dashboard visualizing the statistical results from the queries in the form of treemap, charts, etc. 2) a browser-based generative heatmap of the traffic incident logs [8].

In this system the results of the ESP queries are consumed by a number of visualizations. Fig. 8 shows a Panopticon© dashboard which provides a real-time view of the traffic incidents being reported by geographic area.

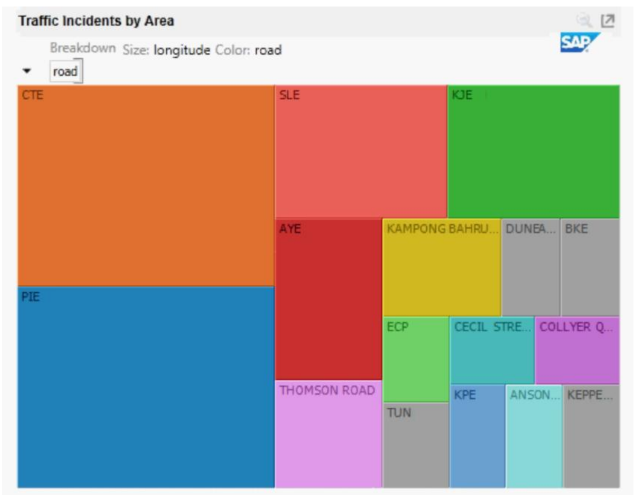


Fig. 8. Real-time treemap visualization of traffic incident logs by area.

Another interesting analytical use-case that we derived was the traffic incidents by type. Fig. 9 shows a real-time pie chart that classifies traffic incidents based on different categories.

One of the useful methods to visualize spatio-temporal data would be a heatmap that has varied intensity based on the occurrence of traffic events. But, the challenge here was to identify a method to account for the constant change in the intensity due to large volumes of the data. For this, we make use of Leaflet, a javascript framework that helps generate a dynamic heatmap layer over OpenStreetMap.

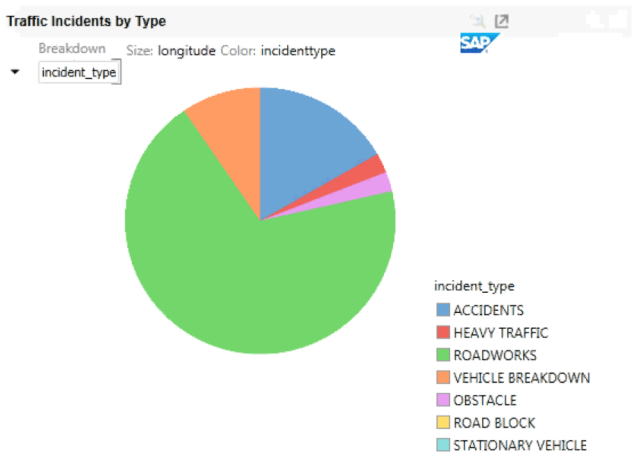


Fig. 9. Real-time pie chart visualization of traffic incident logs by type.

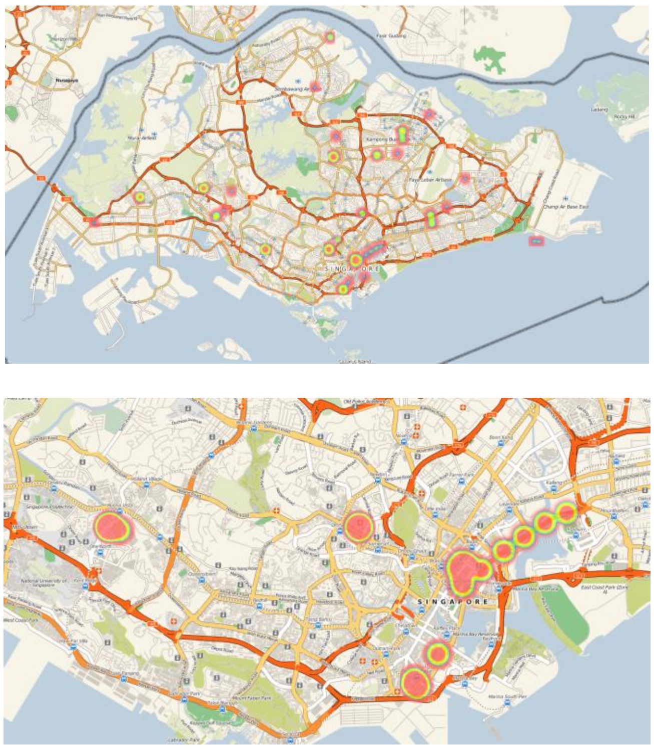


Fig. 10. Snapshots of the real-time heatmap of traffic incident logs.

In Fig. 10 we see the heatmap that illustrates where most of the traffic incidents occur as they are being reported in simulated real-time over the entire day. This can be utilized by operators or daily commuters to make routing decisions.

V. Conclusion

In this paper we have presented a system that can derive real-time analytics over data sources of both high volume and velocity, and also visualize the results in real-time. The traffic and weather logs were used in order to highlight the capability of the system to handle events that can be generated from mobile apps or in general, sensor networks. The analytics derived from the app logs can be extended to other domains that contain similar spatial-temporal data such as weather forecasting, urban transportation, etc. The idea is to extend this system to address 1) multiple data sources 2) providing context based services to users and, 3) develop more intuitive methods of data visualization.

Acknowledgment

We would like to thank buUuk, Singapore for developing TrafficLah and WeatherLah mobile apps and data collection; the Economic Development Board and National Research Foundation of Singapore for supporting this research which can be extended and applied to any generic geo-spatial real-time urban solutions; to the National Environment Agency, Singapore and Newton Circus for organizing the Clean & Green Hackathon 2013.

References

[1] F. Alt, A. S. Shirazi, A. Schmidt, U. Kramer, and Z. Nawaz, “Location-based crowdsourcing: Extending crowdsourcing to the real world,” in *Proc. the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*, 2010, pp. 13-22.

[2] M. Raento, A. Oulasvirta, R. Petit, and H. Toivonen, “ContextPhone: A prototyping platform for context-aware mobile applications,” in *Proc. IEEE Pervasive Computing*, vol. 4, no. 2, pp. 51-59, 2005.

[3] K. S. Hwang and S. B. Cho, “Landmark detection from mobile life log using a modular Bayesian network model,” *Expert Systems with Applications*, vol. 36, no. 10, pp. 12065-12076, December 2009.

[4] Andreescu, M. Paul, and D. B. Frost, “Weather and traffic accidents in Montreal, Canada,” *Climate Research*, vol. 9, pp. 225-230, 1998.

[5] J. Andreyl and R. Olley, “Relationships between weather and road safety: Past and future research directions,” *Climatological Bulletin*, vol. 24, pp. 123–137, 1990.

[6] S. Thordarson and B. Olafsson, "Weather induced road accidents, winter maintenance and user information," *Transport Research Arena Europe 2008*, pp. 72, 2008.

[7] N. Nandan. “Live analytics on high velocity sensor data streams using event-based systems,” *Journal of Industrial and Intelligent Information*, vol. 1, no. 1, pp. 81-85, March 2013.

[8] C. Rohrdantz, D. Oelke, M. Krstajic, and F. Fischer, “Real-time visualization of streaming text data: tasks and challenges,” in *Proc. IEEE VisWeek*, October 2011.

Naveen Nandan is from India and was born on March 06, 1988. He received a Bachelor of Technology in Electronics and Communication Engineering from Amrita Vishwa Vidyapeetham, Coimbatore in 2009. He continued to pursue Master of Science at the School of Computer Engineering, Nanyang Technological University, Singapore where his thesis work was on “Multi-class Emotion Detection from Suicide Notes” applying techniques from natural language processing for feature extraction, association rule mining for feature selection and machine learning for classification.

He started off his career as systems engineer with the Education & Research Department at Infosys, Mysore. He spent a brief stint as systems engineer with JamiQ.com, a social media intelligence startup in Singapore, after which, he worked as software engineer (platform) on the LIVE Singapore! project with the MIT SENSEable City Lab at the Singapore-MIT Alliance for Research and Technology under the Future Urban Mobility program. He currently works for SAP Research and Innovation, Singapore with the Customer Engagement and Strategic Projects Program as research & development engineer. His research interests include distributed & real-time systems, software engineering, complex event processing, data mining, machine learning, natural language processing and data visualization.

Wang Ting was born in 1983 in Chengdu, Sichuan, China. He came to Singapore for his undergraduate studies under Singapore Government-Linked Company SM2 scholarship in 2001. He obtained his Bachelor's degree with honors in 2005 and subsequently Ph. D in 2011 both at Nanyang Technological University (NTU), Singapore.

He worked as a demand planner at Apple South Asia and joined SAP as a data scientist in 2012. His research interests include data mining, mathematical modeling and algorithmic optimization.

Dr. Wang loves soccer. He considers family as his greatest award. He has a son, and he is a good cook — said his wife.

Daniel Dahlmeier was born on January 28, 1982 in Paderborn, Germany. He obtained his Ph.D in Computer Science from the National University of Singapore in 2013 and holds a Diploma in Computer Science from the Karlsruhe Institute of Technology.

He is a researcher with SAP Customer Engagement and Strategic Projects Research and Innovation since 2012. Daniel's research interests are in the areas of machine learning and natural language processing, and how these technologies can create new value from large amounts of data.