dGrav: Visualizing Data Gravity in a Network*

Niño R. Eclarin
Institute of Computer Science
College of Arts and Sciences
University of the Philippines Los Baños
nreclarin@gmail.com

Joseph Anthony C. Hermocilla
Institute of Computer Science
College of Arts and Sciences
University of the Philippines Los Baños
jchermocilla@up.edu.ph

ABSTRACT
Data gravity is a relatively new concept which suggests that data and other entities(applications and services) in a network exert force on each other. dGrav computes and provides visualization for data gravity to allow users to look for patterns and trends in an application's behaviour in a network. Using latency, bandwidth, average request per second, average request size, application mass, and data mass, data gravity between an application and its data can be computed. We evaluated dGrav on a LAN with a shared MySQL database server where applications installed on other servers connect to access data. dGrav automatically retrieves and computes the values by executing database queries and running network tools. Computed data gravity values are stored in a log file and then rendered for visualization.

Categories and Subject Descriptors
C.2.4 [**Computer-Communication Networks**]: Distributed Systems—*Distributed applications, Distributed databases*

General Terms
Management, Measurement, Performance

Keywords
Data Gravity

1. INTRODUCTION
1.1 Background of the Study
Storage and usage of data is one of the most crucial parts of our day to day life. We depend on data in almost every way we can imagine. Different services and applications that we use access specific data in order for them to function. Data is just an abstract value used to represent certain set or sets of items. Think of data as if it were an object that contains certain amount of mass. If the data size increases, so does its mass. This increase will attract applications and services the same way as gravity of a larger object attracts other smaller objects[3]. The earth attracts the moon, at the same time; the moon also attracts the earth. The attraction between the moon and earth is caused by gravity. Think of an application and the data set that it access. If the application only accesses some values in the data set, changes in the data will only have small effects on the application. But if an application accesses several values in the data set, the application becomes highly dependent on the data. This relation between the application and the data set can be viewed as some sort of gravitational force that exists between the two entities.

The applications and services that access data through a network can also be represented as objects with corresponding mass. Data, which has the most mass, becomes the center of the entities. This is for the reason that large data are practically impossible to move. Today, data stored in databases are very large in terms of size and are ranging from gigabytes to terabytes [13]. Large data usually belongs to large companies and high-end service providers. They have accumulated data through the inputs from the applications and services that they provide.

Largely accumulated data is the cradle for fast growing services and applications that exists, even in the Internet. Also, large data dictates where the flow of money is going [11]. Companies and other entities that has stored and accumulated data are the ones who usually dominate their specific playing field. They are the key players that control the flow of money among their designated fields [11]. Accumulative data also dictates how a system or a network behaves. The system or network should adjust to the increasing data size in order to anticipate possible network or system failures. Events due to change in data size can lead to unnecessary repairs if the system or network is damaged. This will increase the cost to maintain the network or system. If data has some sort of gravitational pull among other entities in the network, the relationship between the data and those other entities will also change once the data size is increased.

Data gravity is a new concept proposed by Dave McCrory [3]. In this study, an attempt to visualize the so called "data gravity" will give insights on how data behaves and interacts with other entities around it. This study aims to provide an application, that once installed on a server, or in a given network, will allow the user to have an in depth view of the data

*Available on GitHub at https://github.com/pprmint/spdgrav

gravity. Since this is the first attempt on visualizing data gravity, the application will be designed to handle specific Relational Database Management System (RDBMS).

1.2 Statement of the Problem

The concept of data gravity is a new idea that entities in a network such as applications, services and data exert some sort of pull over each other. This concept affects everyone that uses data stored in a network. As applications access data, the gravity or the attraction of the application to the source of data being accessed affects the way the application behaves. It is important to see the behaviour of the data and the application as they interact on a given network.

A way must be found to see a clear depiction of this interactions between data and other entities in order to know how data gravity may further affect other aspects of the network and its end users. Examining data gravity will involve applications connected through a network. These applications and services will access some data also stored in the specified network. Data size may vary from time to time because the application and services can add or delete data in the network.

Connection speed, network type, and network size must also be considered in measuring the gravity exerted by entities contained in it. A good approach is to develop a mid-ware application that serves as a monitor in between the data source and the application or service that accesses the data.

The application should stand in the middle of the connection to capture both the information from the application and data source. This application will provide a visualization of the data gravity between an application and the data it accesses.

1.3 Significance of the Study

Understanding data and how it behaves is a crucial part in improving services and applications performance. The concept of data having some sort of pull to entities such as applications and services poses the question on the extent of gravity that a certain set of data exerts. If data truly exerts some gravity over entities, visualizing how this gravity works will give a wider scope on how data affects everyday lives. Through this, a vivid image on how applications and services behave in a given network with the influence of the gravity that data exerts can be provided. Because today humans highly depend on networks to communicate and transfer information, visualization on the data gravity that acts on a given network will provide insights on how a certain network grows and interacts with other networks.

This study can also contribute to the tracking of data movements. Learning the implication of a data being moved from one storage or network to another with the influence of data gravity can help us design better networks to facilitate data flow.

Knowing the concept of data movement and gravity exerted by applications and datasets can also help us minimize the cost of building and maintaining networks, content management systems, and data storage systems.

1.4 Objectives

The main objective of this study is to create an application that can monitor the data gravity on a given network, relate the information about the data gravity to other factors specified by the user, and visualize the data gravity in a comprehensive manner that is easily understandable to the end users. The study specifically aims to:

1. build an application that can capture all the necessary information inside the network that can contribute to data gravity;
2. compute the data gravity present among the entities within the network using predefined metrics and standard mathematical formula used to compute general gravity;
3. visualize data gravity using existing tools and present it in a comprehensive and simple way; and
4. provide an interface where the user can manipulate, configure and specify metrics used by the application.

1.5 Scope and Limitation

dGrav uses MySQL as its database. It monitors database accesses to obtain the data mass and data size within a given time interval. Sample data for the applications is generated to simulate large data sets. Existing applications used as projects in previous subjects serve as the applications side of the network. These include past projects in web programming, database, software engineering, and numerical analysis. It uses ping and Iperf[8] to gather network traffic and packet transfer information. Database and application servers need to be installed in different machines. The user needs to grant root access to the application through an interface provided along with the system. Without root access to the machine and MySQL, the application will fail to gather the needed information. Data visualization is handled using Highcharts[7] and Google visualization tools[6]. Other functionality are written using javascript for AJAX and dynamic page loading.

2. REVIEW OF RELATED LITERATURE

The concept of data gravity is just a new idea that arises from the idea that data and other entities in a network can be represented as objects that has mass. There are only a handful of article on the Internet that discuss concepts of data gravity. There is also currently no printed material that discusses data gravity. More so, there are not any types of application that visualizes and conceptualizes the data gravity in a given network.

Data gravity is a term coined by Dave McCrory on one if his blogs last 2010. However, it was not further developed until early 2012. Data gravity is the force of attraction between the data and the application or service that accesses it. McCrory originally illustrated data gravity in cloud products and systems but like the gravity that acts on physical objects, data gravity may also be applicable to other system and network setup [3].

Jack Clark supported McCrory's idea that if a cloud has great amount of data gravity, it would take more effort to

fetch data from it. This follows the concept of gravity. As an object gets closer to the source of gravity, it becomes difficult for the object to escape gravitys effect[12].

Another article that supports data gravity, written by Joe Brockmeier, states that McCrorys theory is very much noticeable when a Content Management System (CMS) migration happens. During a CMS migration, existing data on a content management system is transferred to a different location or storage. Such data transfers also occur in Dopbox, Amazon and iTunes. One symptom of data gravity on these applications is that it is easy for data to get in, but getting it out takes a lot of resources[11]. On a separate article by Jack Clark, he discussed that according to McCrory, it is not only the data that becomes difficult to pull out once the data mass of a cloud increases. This is because greater data mass means greater hold of the customers. Clark also states that companies can use data gravity to look and examine their future prospects [4].

In 2012, McCrory released a new article involving the possible formula for data gravity. This formula is based on the general gravity model. He emphasized that the application mass and the data mass are the key factors on determining the data gravity of a given network. Also, the bandwith, latency and request count in the network significantly contributes to data gravity. Knowing the data gravity will provide relevant information so that a network can be visualized using the gravity of its entities. According to Monserrat, it is easier for the user to view information using their intuition and perception[14]. Visualizing the data or information so is relevant so that the user can interpret the state of a given situation or event. It also shows recognizable patterns, data trends and connections among other data or entity according to Kaidi [5].

3. THEORETICAL FRAMEWORK

3.1 Data Gravity Formula

$$F = \frac{(m_d * m_a) * n}{(l + (\frac{r}{b}))^2} \tag{1}$$

$$m_d = V_d * D_d \tag{2}$$

$$m_a = V_a * D_a \tag{3}$$

$$V_a = memoryUsed + diskUsed \tag{4}$$

$$D_a = cR_{memory} + cR_{disk} + utilization_{total} \tag{5}$$

- F - Force in Megabytes(MB) per square second
- m_d - Data mass (in MB)
- V_d - Data volume (total size of data in MB)
- D_d - Data density (compression ratio of the data)
- m_a - Application mass in MB
- V_a - Application volume (total size of application in MB)
- D_a - Application density (compression ratio of the application)

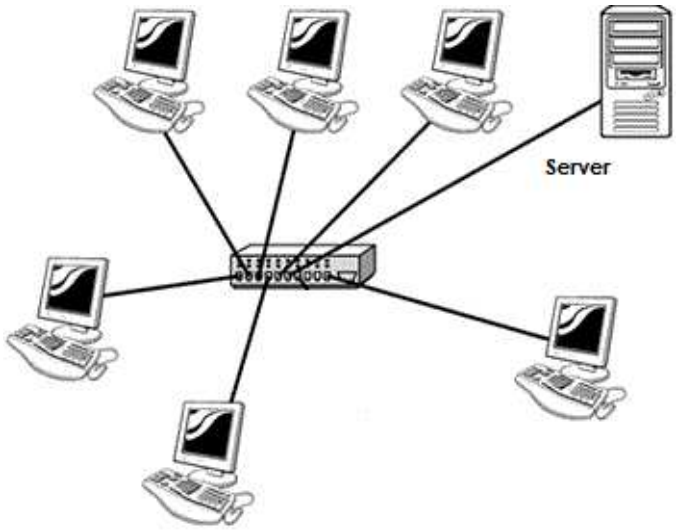

Figure 1: Network topology of the test environment.

- $memoryUsed$ - Amount of memory used by the application
- $diskUsed$ - Disk space used by the application
- cR_{memory} - Compression ratio of memory (in MB)
- cR_{disk} - Compression ratio of disk (in MB)
- $utilization_{total}$ - CPU utilization (in GHz)
- n - Number of requests per second
- l - Latency between the data server and the application server
- r - Average request size in MB
- b - Bandwidth in MB per second between the data server and the application server

McCrory's formula is based on the gravity model of trade formula[1]. This formula involves two entities having mass and the distance between the two entities. Data mass and application mass represent the two entities while the combination of bandwidth, latency, and average request size represent the distance between the two entities [1].

3.2 Network Topology

To simulate a network, application servers are configured in each PC. The PC setup follows a star topology(Figure 1). The server with the data is accessible to the other PC through a switch.

4. METHODOLOGY

4.1 System Requirements

- PCs connected to a LAN
- WAMP Server [10]
- Iperf [8]
- cURL [2]

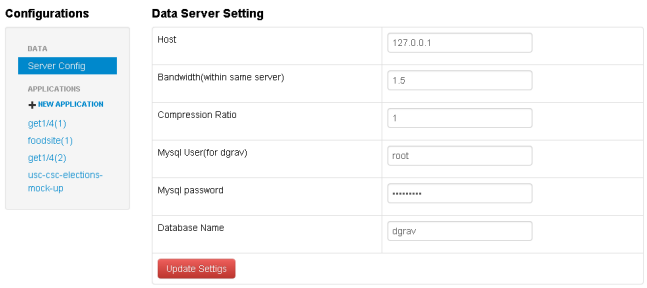

Figure 2: Database server configuration interface.

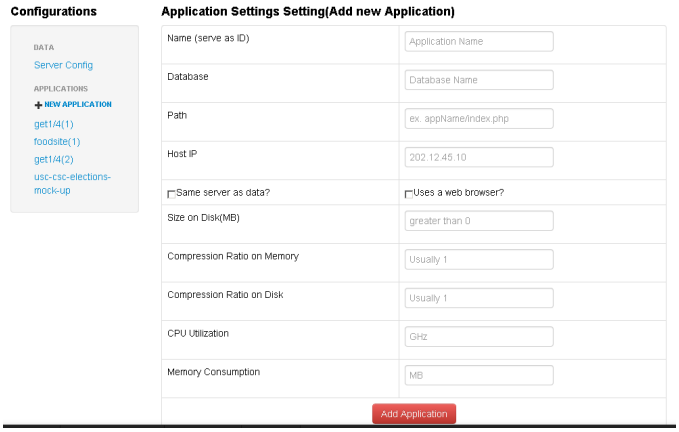

Figure 3: Application configuration interface.

- ping (a built-in operating system tool)
- jQuery 1.7.1 [9]
- Highcharts 3.0 [7]

4.2 Data Samples and Application

After the shared database server PC is configured, applications are installed on the other PCs. Each application connects to the database server to fetch or write data in their respective databases. The database server must also be properly setup. dGrav uses only one table in the database to track the applications which must be installed first before proceeding to application installation. cURL must be installed to allow calling of a PHP script that simulates a cron job. The application must be added and configured using the add application option of dGrav. The fields must be properly filled so that dGrav can monitor the proper application and map it to its corresponding database. Some of the fields include the IP of the application server, the database name and the size of the application. After installing each application on its server, Iperf is installed and run as a background process.

4.3 Formula Modules

To get the data gravity of an application in the network, dGrav uses several computation modules. These modules gather the needed values for the component variables of the data gravity formula. Each module can be configured depending on the platform where it will be deployed. The

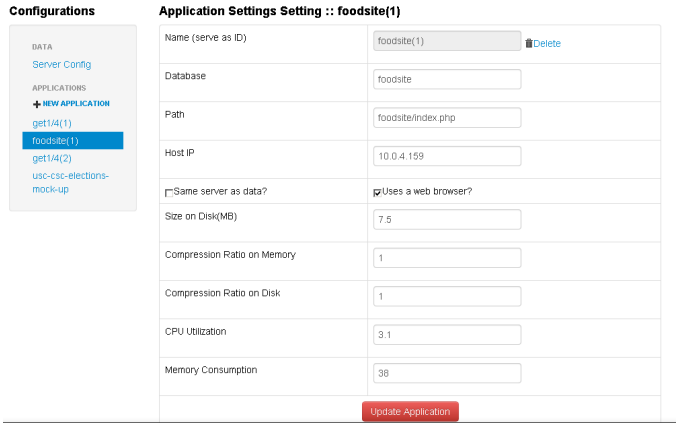

Figure 4: An example configuration for the Foodsite application.

module files are located in the dgrav/core/api/formula modules folder. The following subsections describe the modules.

4.3.1 Data Mass Module

Data mass calculation, computation and gathering is done in MySQL. After configuring the root access for the system, MySQL can now check the total size of the database and its components. These are necessary for getting specific information regarding the state of the database. It outputs information such as the total size of the database, the compression rate and other information need in order to compute for the data mass. The output is stored in a configuration file for future reference and usage.

4.3.2 Application Mass Module

Before application mass can be calculated, the list of applications connecting to the database is listed first(Figures 3 and 4). This is done using the *app* table in dGrav's database. Application information can be edited in the application selection interface. After getting the list of active applications and mapping them to the corresponding MySQL databases, application size and memory usage is computed based on each *appconfig* file located on their corresponding folders. These folders are auto generated once the user adds a new application. The output is stored in a text file for future reference and usage. This process runs simultaneously with the PHP scripts that run the MySQL queries.

4.3.3 Latency Module

Latency is computed using the ping command. The database server pings the application server and stores the result on a file. Latency value is in seconds.

4.3.4 Bandwidth Module

Bandwidth is computed using Iperf. It is a tool to dynamically get the bandwidth between two nodes in a network. Each application server has an Iperf server running in them. The database server uses Iperf client and sends requests to each Iperf server in order to retrieve the bandwidth between it and the application servers. The bandwidth is represented in MBps.

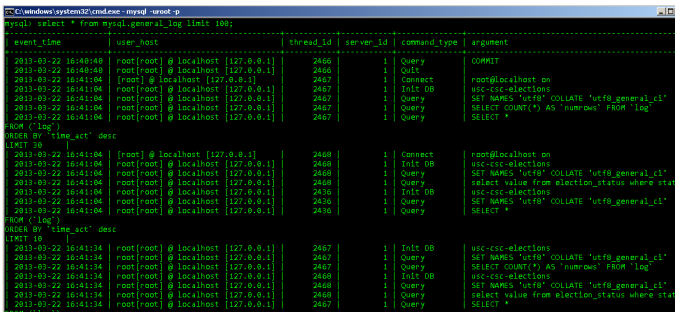

Figure 5: Contents of MySQL general log.

4.3.5 *Request-handling Module*

Request-handling module returns both values for average request per second and average request size. Request module utilizes the MySQL General log(see MySQL documentation for details). MySQL general logs records information about a MySQL process such as time, process type, database involved and the query itself. This module uses the unique IP of the application server and its unique database to map each MySQL general log entries to its corresponding application. The result sample(Figure 5) is parsed and the query for each specific application is selected. The size of the queries are added and returned as the average request size(MB). The time interval is checked and then divided by the total request counted for that time interval. This returns the average request per second.

4.4 Data Gravity Computation

After all the necessary information is gathered, a PHP script reads all the files where the information is stored. Each application connecting or accessing the database have its data gravity computed and stored in a log file. Computation of data gravity will be based on the formula proposed by McCrory. Once all the computation is done, the application logs the current state of the network and all the application connected in it. The application will run a computation based on the interval specified by the cron task. Once a full record has been done, the application clears out the MySQL general log table to facilitate future storing of data.

4.5 Visualization

dGrav uses several graphing and chart tools to render its visualization components. One extension is Highcharts(Figure 6). It is a javascript tool which renders different kinds of graphs depending on the type of data to be interpreted. Another tool is Google visualization tools(Figure 7). It uses motion graphing styles to show relevant information and trend pattern on data.

There are also visualization options that the user can select. The user can compare different variables(up to five variables) to better understand relations and trend patterns among data gravity and network components(Figure 9). Also, there are guides and help sections that can help the user interpret the data.

4.6 System Configuration and Admin Management

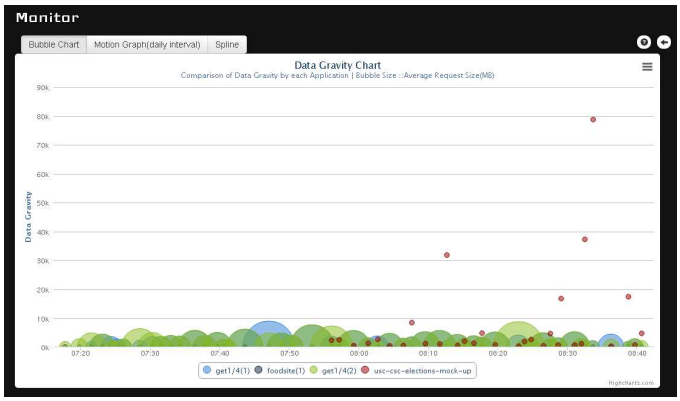

Figure 6: A visualization of data gravity across a time interval.

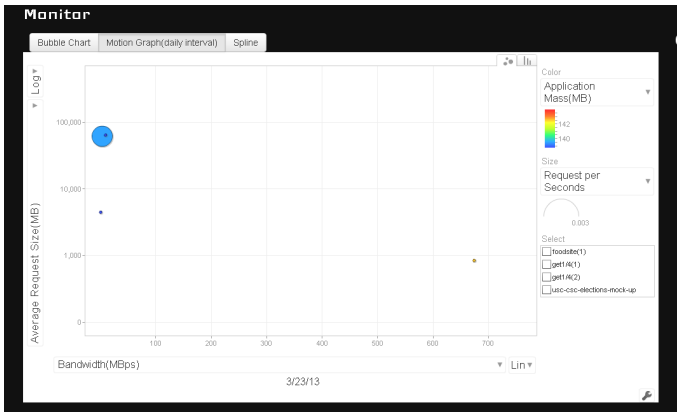

Figure 7: A visualization of data gravity in relation to the component variables.

When installing the system into the server, a configuration page is provided to the user. This configuration page will allow the user to grant root access to the database and the operating system. Root access is essential for some system tools will not run if it does not have root permission. After proper configuration, an interface where the Admin of the system can view the status of each application and data server is now available.

5. RESULTS AND DISCUSSION

5.1 Data Gravity Component Values Retrieval

Values needed for data gravity involves access to MySQL requests. A look at the formula for data gravity shows that it is highly dependent on the number of requests per second (n). This factor multiplies the application and data masses which are also large contributors to data gravity.

Data mass and application mass are proposed formula that can be justified by its basis to the general gravity formula. The masses needed both density and volume, both of which are clearly justified in the theoretical framework.

Average requests and average requests size however is unpredictable and is difficult to asses due to the many factors affecting its numeric representation. These statements

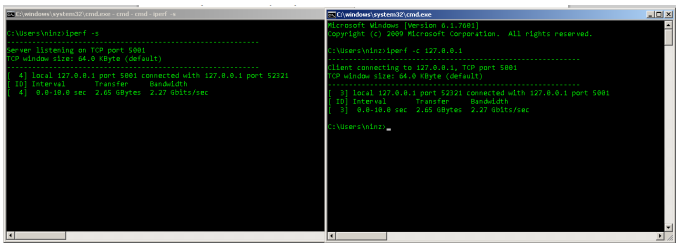

Figure 8: Iperf server(left window) and Iperf client (right window).

Figure 9: Users can select the component variable to display with data gravity.

are based on the observation that both the average requests and average request sizes can change its value based on the platform the network is deployed. If the user uses PostgreSQL as its DBMS, the formatting and logging of queries can change to PostgreSQL standards. Also we are not just talking about DBMS type of platform. What if the request representation is from a client to an application server, not a database server. The request format then would be the total size of the POST or GET requests that the client sends to the application. In this case, the server logs will be the one responsible for the values needed for the module.

5.2 Visualization

Using Highcharts and Google visualization tools, dGrav has represented data gravity and its components. Several options has been made available to the users such as comparison between custom variables, comparison between applications and transits between rise and fall of data gravity.

One feature allows the comparison of data gravity with respect to a selected variable. The bubble chart(Figure 9) shows that the user can select which parameter the size of the bubble will be based.

Application selection and comparison is also possible. The bottom panel of each visualization provides a select option for the variables to compare. The user can remove or select what to be shown in the graph. Another feature is the selection and breakdown of data gravity component variables per application. The user can select variables and examine its relation to data gravity and other variables (Figure 10).

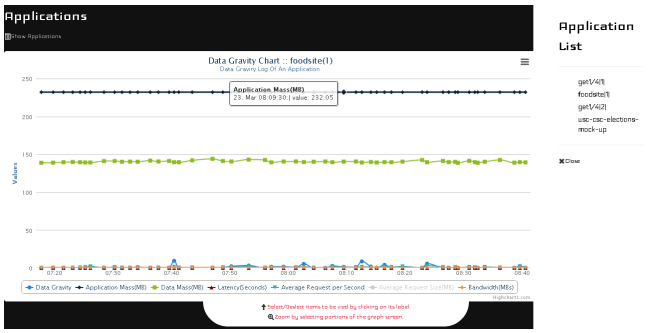

Figure 10: Variable breakdown per application.

5.3 Data Interpretation

Data gravity results from the pull of data from the data server allowing other applications and services to stay by its perimeter. Different factors such as request size and frequency may contribute greatly to data gravity. By the assumption that data gravity affects the dependencies of applications and services, one can observe that high data gravity means that data migration can be difficult at times.

Based on the nature of the applications tested by dGrav, high *read* applications, or applications that just retrieve data from the server tends to have lower data gravity. This is because *read* applications just display the data it gets from the server. There are no modifications or other operations that may cause the nature of data to change. In short, such applications have low footprint or does not tend to affect other entities around it. This observation might change if a different setup is used.

On the other hand, high *write* applications have higher data gravity because it changes the nature of data. It may add large amount of data in the data server at any time. When it add or manipulates data, the changes might also affect other entities in the network. If the application with high data gravity needs to be migrated to other data servers, it might be difficult to do so because there are so many dependencies that the application needs to satisfy first. Also, if an application has high data gravity, the services and other applications dependent on it might have problems if that application is to be changed. This pattern is visible on the current setup but might vary between platforms and applications. It also depends on the services provided by the application. dGrav provides the tool and other module but the data still needs to be analyzed and interpreted properly.

The observation and other data patterns can be interpreted in many ways. The tools that dGrav provides allow application statistics to be viewed in comparison to other applications. Also the trend in data is only observed in a small set test environment. Although the data set is small, it is big enough to simulate some results and show pattern based on the simulated architecture and nature of the applications installed.

6. CONCLUSION

Data gravity can be represented and visualized using different tools. The effect of data gravity on a network and

some of its aspects is still a puzzle that needs to be put into pieces. Through dGrav, some aspects have been brought to light. Data gravity has some effect on the application behaviour and the services on a given network. Once we learn these effects, patterns and trends we might be able to predict application crashes, network maintenance, data migration patterns and improvements and other data-related occurrences within a network.

7. ACKOWLEDGEMENTS

The authors would like to thank the reviewers for their comments.

8. REFERENCES

[1] A Formula for Data Gravity. http://datagravity.org/2012/06/26/a-formula-for-data-gravity/.
[2] cURL. http://curl.haxx.se/.
[3] Data Gravity in the Clouds. http://blog.mccrory.me/2010/12/07/data-gravity-in-the-clouds/.
[4] Data gravity: The reason for a cloud's success. http://www.zdnet.com/data-gravity-the-reason-for-a-clouds-success-4010026102/.
[5] Data Visualization. http://www.cs.uic.edu/~kzhao/Papers/00_course_Data_visualization.pdf.
[6] Google Charts. https://developers.google.com/chart/.
[7] HighCharts. http://www.highcharts.com/.
[8] Iperf. https://iperf.fr/.
[9] jQuery. https://jquery.com/.
[10] WAMP Server. http://www.wampserver.com/en/.
[11] What "Data Gravity" Means to Your Data. http://www.readwriteweb.com/cloud/2012/05/what-data-gravity-means-to-your-data.php.
[12] Will you get locked into your cloud? Ask the data gravity theory. http://www.zdnet.com/will-you-get-locked-into-your-cloud-ask-the-data-gravity-theory-7000000133/.
[13] M. Bacman. *Star Author Database Implementation in Java and Java Server Pages*. 2002.
[14] T. Monserrat and J. P. Pabico. Nettrace: Information visualization of a website using quasi-hierarchical graphs in 3-d hyperbolic space. In *ICS Technical Reports*, 2008.