Joseph Anthony C. Hermocilla

Full papers: http://jachermocilla.org/publications

KIMIKA Volume 33, Number 1, pp. 09-19 (2022)
 ISSN 2508-0911
https://doi.org/10.26534/kimika.v33i1.09-19

Molecular Docking of Ecdysone Agonists and Limonoid Components of Calamondin Seeds to Tobacco Budworm Ecdysone Receptor

Annabelle T. Abrera[1,2], Melvin A. Castrosanto[2], Joseph Anthony C. Hermocilla[1,3], Marlon N. Manalo[1,2]*

[1] *Computational Interdisciplinary Research Laboratories, University of the Philippines Los Baños, College, Laguna 4031*
[2] *Institute of Chemistry, University of the Philippines Los Baños, College, Laguna 4031*
[3] *Institute of Computer Science, University of the Philippines Los Baños, College, Laguna 4031*

*Author to whom correspondence should be addressed; email: mnmanalo@up.edu.ph

ABSTRACT

Molecular docking of calamondin limonoids to tobacco budworm ecdysone receptor was performed to calculate binding affinities and evaluate their potential as an insecticide. Results were compared to those of 20-hydroxyecdysone, ponasterone A, and known ecdysone agonists by examination of binding interactions between the ligands and receptor. While an earlier study by other researchers attributed the stronger binding of ponasterone A to (de)solvation effects, molecular dynamics simulations in this study revealed stronger hydrogen bonds between the receptor and ponasterone A, consistent with calculated binding affinities from molecular docking. None of the limonoids and ecdysone agonists showed stronger binding affinity than 20-hydroxyecdysone, suggesting that site-specific docking may not be suitable to predict the binding interactions of these non-steroidal ligands with tobacco budworm ecdysone receptor. The computational approach described in this study can be used in the preliminary screening of compounds for bioactivity against receptors with known active sites.

Keywords: *ecdysone receptor; limonoid; tobacco budworm*

INTRODUCTION

Due to the persistence and harmful effects of several insecticides available in the market, many studies are now focused on developing environmentally benign insecticides. One such group of compounds that have been reported to have potent effects on insects and low toxicity to non-target organisms are limonoids (Abdelgaleil and El-Aswad, 2005). Limonoids are the prominent secondary metabolite present in citrus and are responsible for the bitter taste in juice. Limonoids are described as modified triterpenes derived from a precursor with a 4,4,8-trimethyl-17-

Development of a real-time wireless sensor network-based information system for efficient irrigation of upland and lowland crop production systems

R C Ramirez [1,*]***, E S Agulto***[1]***, S D Glaser***[2]***, Z Zhang***[2]***, J A C Hermocilla***[3] ***and V B Ella***[1,*]

[1] Land and Water Resources Engineering Division, Institute of Agricultural and Biosystems Engineering, College of Engineering and Agro-Industrial Technology, University of the Philippines Los Baños, College, Laguna, 1430, Philippines
[2] Department of Civil and Environmental Engineering, University of California, Berkeley, California, 94720, USA
3 Institute of Computer Science, College of Arts and Sciences, University of the Philippines Los Baños, College, Laguna, 1430, Philippines

[*]Corresponding authors: Engr. Ruzell C. Ramirez, rcramirez6@up.edu.ph; Dr. Victor B. Ella, vbella@up.edu.ph

Abstract. This study developed a real-time web-and WSN-based information system for efficient irrigation water management and automation of drip-irrigated upland crop and intermittently-irrigated lowland crop production systems. The web-based system uses Flutter and DART to accommodate multiple end user platforms, while the WSN-based system uses state-of-the-art hardware and sensors for real-time monitoring of soil moisture, water level and weather conditions. The sensors are wirelessly connected in a low-power mesh network that sends data to a central server. The sensor readings are uploaded to the web application via MQTT, which generates charts and graphs for data analysis. The sensor readings compared well with measurements from conventional instruments. The system in this study provides a sustainable solution for improving irrigation efficiencies under both upland and lowland crop production systems, in minimizing water losses and in improving the overall agricultural crop productivity.

1. Introduction

The Philippines is home to large areas of agricultural land. In fact, within its 30 million hectares of land area, 44% is agricultural [15]. These wide harvest areas are used to grow upland and lowland crops. But despite the size of the country's agricultural real-estate, most of the farmers in the Philippines, especially in the small to medium scale, are stuck using traditional methods of irrigation and water management [16]. This is highly inefficient because these methods often prioritize immediate gain more than overall sustainability. It also leads to high labor and water consumption costs. It can also result in reduced crop yield and crop production because of unintentional under or over-irrigation. For instance, in 2012, 16% percent of the country's rice output was lost due to inefficient farming methods and equipment [17].

Pursuant to the Department of Agriculture's vision of modernizing the country's agricultural sector and to address issues like climate change, climate variability, aging farmers, inefficient irrigation systems among others, this study was conceptualized to develop a smart irrigation system. In general, smart irrigation systems help improve irrigation efficiency and agricultural sustainability by using wireless sensor networks (WSN) that transmit sensor parameters such as volumetric soil water content, temperature, and humidity to information systems, which then serve as basis for irrigation scheduling. These systems allow farmers to monitor and automate irrigation of crops. Furthermore, the system can

Characterization and Classification of Malware Traffic over the Tor Network

Marie Betel B. de Robles, Joseph Anthony C. Hermocilla, and Jaderick P. Pabico
Institute of Computer Science, College of Arts and Sciences
University of the Philippines Los Baños, College, Laguna
{mbderobles2,jchermocilla,jppabico}@up.edu.ph

ABSTRACT

Tor is a popular anonymity tool for digital-based communication that is used to protect the users' identity and is also utilized to avoid any eavesdropping and man-in-the-middle attacks. It implements the concept of onion routing where traffic is being routed to relay nodes which hide users' identity and secure the data being transferred over the internet. As its use increases among internet users, there is a need to monitor the traffic within Tor to ensure that it is not being misused. There are a number of ways Tor is used for malicious purposes, one of which is to hide malware traffic. Since Tor traffic is encrypted, traditional approaches such as port examination and packet inspection are ineffective. In this study, the automatic classification of Tor web traffic into malware and non-malware types was conducted using machine learning approaches. A dataset of positive and negative malware web traffic examples was generated from VirusTotal's malware applications and regular web traffic. The dataset was generated from a controlled network environment using an automated system that was developed in this work. Various machine learning (ML) techniques were employed to evaluate their respective effectiveness in classifying between malware and non-malware Tor traffic. The ML considered were C4.5 variant of decision tree, K-nearest neighbors, Naive Bayes, and Random Forest. The ML respective effectiveness in classification were evaluated against the classification metrics Precision, Recall, False Positive Rate, and False Negative Rate using a standard statistical analysis of variance (ANOV) procedure. ANOV results show that the MLs respective performances in evaluating the metrics are not statistically different from each other at a significance of $\alpha = 0.05$.

CCS CONCEPTS

• **Security and privacy → Network security**;

KEYWORDS

Malware, Tor, Flows, Machine Learning

ACM Reference Format:
Marie Betel B. de Robles, Joseph Anthony C. Hermocilla, and Jaderick P. Pabico. 2020. Characterization and Classification of Malware Traffic over the Tor Network. In *Proceedings of Philippine Computing Science Congress (PCSC2020)*. Baguio City, Philippines, 10 pages.

PCSC2020, January 2020, Baguio City Philippines

1 INTRODUCTION

The internet has been continuously evolving and expanding its reach to people's everyday lives. Today, users can access new ideas and information and perform activities because innovations in communication, information sharing, education, and other related technologies can be easily utilized. As the use of internet is increasing, anonymity while on it became one of the most appealing services for some users. Encrypting information about users' activities allows users to exercise freedom of speech, avoid network surveillance, and seek discomfiting information. There are many ways for users to obtain anonymity on the internet, one of which is through Tor.

Tor is a popular anonymous proxy tool that is being used by millions of users worldwide. It protects users' data using the onion routing concept where data is wrapped in multiple layers of encryption. As it was originally developed for protecting government communications, it is now being used by a wide range of users such as journalists, activists, business executives, and others who prefer to protect their identities while exchanging data and communicating over the internet. It also hosts hidden services accessible only through the Tor network itself [3].

Users with malicious intent have also become attracted to anonymity tools like Tor. Today, numerous unlawful services and activities are being conducted over an encrypted network. Even malware has also utilized encryption for its communication [8]. In the study by Anderson et al. [6], about 10% of their collected malware communication samples were encrypted. For example, the malware Locky is a ransomware that prompts the victim to visit a Tor hidden service to pay the ransom and retrieve stolen files [7]. Since Tor can be used by anyone, attackers can easily hide any malware tracks through it. This threat makes it important to monitor Tor traffic to determine if its use is for malicious purposes.

Traditional traffic classification approaches such as Deep Packet Inspection (DPI) and signature-matching are ineffective over encrypted traffic. Today, machine learning (ML) techniques are being applied where the features used are derived from statistical characteristics of internet traffic [23]. Certain classes of applications hold these unique features and are used in ML algorithms for traffic classification. This study aims to classify Tor web traffic into malware and non-malware types and to develop a computational workflow that can classify malware traffic over the Tor network through ML approaches.

2 RELATED WORK

Several works have already been done to analyze Tor traffic but only a few studies in malware detection.

Design and Implementation of an Autonomous Surface Vehicle Platform for Water Quality Monitoring

Miyah D. Queliste, Joseph Anthony C. Hermocilla and Jaderick P. Pabico
Institute of Computer Science
University of the Philippines Los Baños
(mdqueliste,jchermocilla,jppabico)@up.edu.ph

ABSTRACT

Autonomous surface vehicles (ASV) are water surface robots that move independently without manual operation. They are becoming popular for various applications including water quality monitoring (WQM). The Philippines, though heavily dependent on its water resources, has limited adaptation of ASV technologies for automated WQM. Manual WQM, currently practiced by experts via manned vessel operations or fixed sensors, can be labor-intensive and time-consuming. It also does not provide real-time information to stakeholders. This work presents the design and implementation of an ASV platform for WQM. The platform includes a 0.8 m catamaran-style boat and a software system consisting of a web application and mission scripts. The boat was built with an on-board computer equipped with navigation and control sensors. Water quality sensors like temperature and dissolved oxygen are integrated on the boat for *in situ* data collection. The web application is used to create autonomous missions. Mobile applications were also developed to facilitate the transmission of water quality data to a database. Simulation and field tests showed successful autonomous navigation through predefined mission waypoints and sensing of sound water quality data. Loiter mode was found effective in enabling the ASV to hold position for up to one meter radius.

CCS CONCEPTS

• **Applied computing → Computers in other domains**;

KEYWORDS

autonomous surface vehicles, water quality monitoring, navigation, sensors

ACM Reference Format:
Miyah D. Queliste, Joseph Anthony C. Hermocilla and Jaderick P. Pabico. 2020. Design and Implementation of an Autonomous Surface Vehicle Platform for Water Quality Monitoring. In *Proceedings of Philippine Computing Science Congress (PCSC2020)*. Baguio City, Philippines, 12 pages.

1 INTRODUCTION

Advancements in remote sensing technologies, wireless networks, navigation, and localization systems paved the way towards rapid developments in the area of robotics. Global positioning devices and other primary components like sensors have become more compact and inexpensive. Wireless data systems have also supported longer range and higher bandwidth of communications [14]. Coupling these with autonomous vehicles have offered innovative and vast capabilities that are currently explored worldwide [5, 8, 28].

Autonomous Surface Vehicles (ASV), also called Unmanned Surface Vessels (USV), are water surface robots that move independently without continuous human intervention. These vehicles, without an operator on board, allow other modes of control such as semiautonomous control, tele-operated control, or remote control [9]. They come in various shapes and sizes and are becoming popular for a wide range of research and military applications such as security, defense, bathymetric mapping, environmental sampling, flood response, water monitoring, and agriculture [6, 15].

Apart from bathymetric surveys where ASVs were initially applied in the 1990s, water quality monitoring is also prevalent among available studies in the field. Sensing surface water quality is of increasing interest to researchers as it encompasses several environmental and economic concerns. Data collected from these activities give insights to the suitability of water for domestic use, biodiversity, and the water's ability in supporting aquatic life [11]. Water quality monitoring, for instance, helps in understanding and modeling water contaminants like chemical spills and growth of harmful algal blooms (HABs) [8].

The Philippines, one of the largest fish producers in the world, is heavily dependent on its water resources for economic sustainability. More than 800 thousand Filipinos are involved in the fishing industry in various production levels—municipal, commercial, and aquaculture [23]. The country's lakes are of great interest to water quality research because of the fish kill occurrences in the area. In 2011, more than PHP 57M worth of 700 metric tons of farmed fish were killed in Lake Taal. One identified cause was the sudden rise in temperature followed by rain which resulted in inadequate oxygen in the water surface that suffocated the fish [22, 27]. Other researchers found that it was due to "lake overturn" that pushed up the bottom stratum of the lake and pushed down the upper stratum caused by the changing of the general wind direction. Being a caldera of past volcanic eruptions, the bottom stratum is warm with sulfuric emissions from the lake floor's volcanic vents bringing along with it water layers with practically zero dissolved oxygen. Had there been real-time WQM, the Taal fisher folks would have been given early warning to avoid heavy financial losses [12, 16–18]. Thus, water quality monitoring is of scientific, economic, and governance importance for the country.

PCSC2020, March 2020, Baguio City, Philippines

A MEMETIC ALGORITHM FOR GENERATING SPECTRAL INDICES FOR REMOTELY SENSED IMAGERY

Ivan Marc H. Escamos (1), Maria Art Antonette D. Clariño (1), Joseph Anthony C. Hermocilla (1), Jaderick P. Pabico (1), Cristino L. Tiburan, Jr. (1)

[1] University of the Philippines Los Baños, Los Baños, Laguna, 4031, Philippines
Email: ivanhescamos@gmail.com; mdclarino@up.edu.ph; jchermocilla@up.edu.ph; jppabico@up.edu.ph; cltiburan@up.edu.ph

KEY WORDS: Spectral Index, Memetic Computing, Search Algorithms, Feature Extraction

ABSTRACT: Spectral indices are formulas that integrate different wavelengths, or bands, of the electromagnetic spectrum to accentuate the abundance of various features of interest such as vegetation, burned areas, man-made, water, and geological features. Given its numerous applications, research on the development of spectral indices is a worthwhile undertaking. New spectral indices could be created as improvements over existing ones in aspects such as separability and sensitivity. New spectral indices can be developed for other features not yet covered by any spectral index.

The problem of developing spectral indices was posed as a search problem. The search space consists of all possible spectral indices. A spectral index was viewed as a combination of various spectral bands, mathematical operators, and numerical coefficients. Therefore, search algorithms could be used to generate spectral indices.

For this study, the memetic algorithm was utilized. Specifically, a memetic variant of genetic programming was developed. This was achieved by augmenting the genetic programming algorithm with the simulated annealing algorithm.

The developed algorithm was used to generate spectral indices for vegetation and built-up areas. Training points derived from Landsat 8 imagery was used as the input. The quality of the generated spectral indices was measured using two metrics: Silhouette Score and the Jeffries-Matusita distance. Indices for vegetation and built-up areas were developed: namely Memetic Genetic Programming Vegetation Index (MGPVI) and Memetic Genetic Programming Built-up Index (MGPBI). MGPVI was compared with other vegetation indices and MGPBI was compared with other built-up indices. Both generated indices outperformed their competing indices in terms of the earlier mentioned metrics.

1. INTRODUCTION

An index is a metric that describes complex phenomena using a limited number of parameters. (Bouzekri et al., 2015) In remote sensing, a spectral index is a formula that integrates different wavelengths, or bands, of the electromagnetic spectrum to accentuate the abundance of various features of interest such as vegetation, burned areas, man-made, water, and geological features. (Harris Geospatial Solutions, 2018) Spectral indices were also used by researchers as tools in aiding land cover classification in remotely sensed imagery. (Jabloun et al., 2009; Bhatt et al., 2018)

One of the most popular spectral indices is the vegetation index Normalized Difference Vegetation Index (NDVI) which was developed by Rouse. (Rouse et al., 1974). This index highlights areas of healthy, green vegetation. NDVI utilizes the near-infrared band (0.6 µm) and the visible red band (0.9 µm) and is given by the formula:

$$\mathrm{NDVI} = \frac{\mathrm{NIR} - \mathrm{RED}}{\mathrm{NIR} + \mathrm{RED}} \quad (1)$$

where NIR represents the Near-Infrared band, and RED represents the red band. The RED band is used since healthy vegetation absorbs waves of this frequency well. On the other hand, the NIR band is used since healthy vegetation reflects this frequency well. (Perez and Comiso, 2015)

Given the numerous applications of spectral indices, research on this area is a worthwhile undertaking. New spectral indices could be created as improvements over existing ones in aspects such as separability and sensitivity. Spectral indices can be developed for other features not yet covered by any spectral index. Also, spectral indices could also be devised as feature extraction tools to improve the results of land cover classification

SkyLab : An extensible workflow web application for HPC on the cloud

Vincent Paul L. Carpio
Institute of Computer Science
College of Arts and Sciences
University of the Philippines Los Banos
vlcarpio@up.edu.ph

Katrina Joy M. Abriol-Santos
Institute of Computer Science
College of Arts and Sciences
University of the Philippines Los Banos
khmagno@up.edu.ph

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

ABSTRACT

Most scientific applications require high performance computing (HPC) which utilizes parallel processing to run tasks quickly and efficiently. MPI clusters are often used to cater this type of tasks but the hardware required can be costly. Peak-Two Cloud (P2C) can host HPC applications in a cloud environment which is relatively cheaper due to resource reuse and more convenient due to on-demand provisioning. One of the key features of P2C is vCluster, a tool that can deploy MPI clusters usable through the command line. In this paper, we present the design, implementation, and user-evaluation results of SkyLab, a workflow web application on top of vCluster to simplify the process of running MPI applications for users not accustomed to the command line. SkyLab currently supports applications used in bioinformatics, molecular dynamics, molecular docking, and quantum chemistry. The extensible design of SkyLab enables additional tools to be incorporated easily as modules.

CCS CONCEPTS

•Information systems → Computing platforms; •Networks → Cloud computing;

KEYWORDS

high performance computing, workflows

ACM Reference format:
Vincent Paul L. Carpio, Katrina Joy M. Abriol-Santos, and Joseph Anthony C. Hermocilla. 2017. SkyLab : An extensible workflow web application for HPC on the cloud. In *Proceedings of Philippine Computing Science Congress, Cebu City, Philippines, March 2017 (PCSC2017),* 6 pages.
DOI:

1 INTRODUCTION

Cloud computing has marked significant developments and possibilities in the industry. It focuses on offering services for the different needs of the modern society.

There are three categories of cloud computing services namely, Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). Organizations provide SaaS depending on the demand. Google Apps is one example of SaaS that can be used to manage email and create documents, etc. PaaS offers developers a platform where they can build and deploy applications. IaaS provides storage, servers and clusters. These tools are primarily created to serve computational needs [1]. A cloud computing platform dynamically allocates, configures, reconfigures and deallocates servers as requested or on demand. This approach ensures the elasticity of cloud computing [3].

Most scientific applications require high performance computing (HPC) which needs CPU intensive computations and large data storage. To be able to host such applications, several computers interconnected in a network such as clusters are needed. This makes scientific computing very costly in terms of hardware infrastructure investment. With the advancement in cloud computing, these scientific applications can be deployed in the cloud without worrying about hardware costs and maintenance [1]. However, studies have shown that network transmission delay is the major drawback in deploying HPC applications in the cloud[3].

Peak-Two Cloud (P2C) is a private cloud based on OpenStack designed for research in deploying HPC applications in the cloud[8]. One of the features introduced by P2C is vCluster. vCluster is a tool that enables a user to deploy a working (Message Passing Interface) cluster on demand and to terminate it after use. P2C has been used by researchers in various fields including bioinformatics, quantum chemistry, and molecular dynamics. These researchers belong to different research groups who have little or no investment in HPC infrastructure due to limited funding but requires heavy computing resources for their research. vCluster, however, is a command line application which make it difficult for non-technical users (physicists, chemists, and biologists) to use. A more user-friendly interface is needed in order to enable scientists to focus more on their science rather than on learning and using the command line.

Presented in this paper is SkyLab[1], a workflow web application on top of vCluster that addresses the concern above. Specifically, SkyLab will

(1) allow users to execute HPC tools via web interface;
(2) enable developers to easily extend it to support additional HPC tools;
(3) enable users to share their instantiated clusters; and
(4) support displaying of results using third party tools.

The following are the tools that are currently supported by SkyLab. These are commonly used by collaborators from different research groups.

- *AutoDock* - A software used to simulate protein-ligand docking[13].

[1]https://github.com/vincentpaul12/SkyLab

Benchmarking P2C for HPC using NPB

Joseph Anthony C. Hermocilla

Abstract

We report some benchmarking results of the Peak-Two Cloud(P2C) for High-Performance Computing(HPC) using the NAS Parallel Benchmarks(NPB).

1 Introduction

In order to evaluate the performance of the P2C[1] for HPC, NPB[1] version 3.3.1 was run on a 16-node MPI cluster provisioned using vcluster[2]. vcluster provisions nodes with one core, 1GB RAM, and 20GB disk.

NPB consists of a set of programs that implements different computational approaches associated with Computational Fluid Dynamics(CFD). These programs represent the types of applications that are run in supercomputers and HPC clusters. In running the benchmark, classes A and B were used for each program. Different classes have different problem sizes and parameters which result to different measurements.

The programs were run using 1, 2, 4, 8, and 16 nodes. The completion times were recorded and plotted using gnuplot. Figure 2 shows the hosts where the VMs used in the cluster were instantiated. The state of the hypervisors are shown in Figure 1.

P2C uses the default FilterScheduler of OpenStack. Observe that the host cinterlabs-02 in Figure 1 is overprovisioned. It only has a total of four physical cores but five VCPUs are allocated. In Figure 2, the cluster nodes benchmark-slave-5,benchmark-slave-13, and benchmark-slave-14 are assigned to the host cinterlabs-02.

For related work, see [2].

Hypervisors

Hostname	Type	VCPUs (used)	VCPUs (total)	RAM (used)	RAM (total)	Storage (used)	Storage (total)	Instances
cinterlabs-01	QEMU	4	4	4GB	3.8GB	63GB	454GB	3
cinterlabs-04	QEMU	1	4	1.5GB	3.8GB	21GB	454GB	1
cinterlabs-05	QEMU	2	4	2.5GB	3.8GB	21GB	454GB	1
cinterlabs-02	QEMU	5	4	5GB	3.8GB	105GB	454GB	5
cinterlabs-03	QEMU	3	4	3.5GB	3.8GB	63GB	454GB	3
cinterlabs-06	QEMU	1	4	1GB	3.8GB	21GB	454GB	1
cinterlabs-07	QEMU	4	4	4GB	3.8GB	84GB	454GB	4
cinterlabs-08	QEMU	3	4	3.5GB	3.8GB	63GB	454GB	3
cinterlabs-09	QEMU	4	4	4GB	3.8GB	84GB	454GB	4
cinterlabs-11	QEMU	3	4	3.5GB	3.8GB	42GB	454GB	2
cinterlabs-10	QEMU	3	4	3.5GB	3.8GB	63GB	454GB	3
cinterlabs-12	QEMU	3	4	3.5GB	3.8GB	42GB	454GB	2

Figure 1: Hypervisor state after the 16-node cluster was created.

[1] https://www.nas.nasa.gov/publications/npb.html
[2] http://srg.ics.uplb.edu.ph/projects/peak-two-cloud/peak-two-cloud-resources/deployinganmpiclusterusingvcluster

Exploring RSSI for UAV-to-GCS distance estimation

Guia Carmella B. Lara
Institute of Computer Science
College of Arts and Sciences
University of the Philippines
Los Banos
gblara@up.edu.ph

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

Danilo J. Mercado
Institute of Computer Science
College of Arts and Sciences
University of the Philippines
Los Banos
djmercado@up.edu.ph

ABSTRACT

This study investigated the use of the Received Signal Strength Indicator (RSSI) to improve the Follow Me autonomous flight mode of an Unmanned Aerial Vehicle (UAV). RSSI data was fused with Global Positioning System (GPS) information to estimate the distance between the mobile device Ground Control Station (GCS) and the UAV. Using a weighted scheme, the estimated distance was computed and used to adjust the position of the UAV with respect to the GCS. Several schemes were tried: GPS only, combinations of GPS and RSSI, and RSSI only. Using GPS only gave the lowest mean distance error, but certain combinations of RSSI and GPS gave lower variances than GPS only.

CCS CONCEPTS

• **Computer systems organization** → **Robotics**; **Robotic autonomy**; • **Networks** → *Wireless access networks*;

KEYWORDS

UAV, Drones, distance estimation, RSSI

ACM Reference format:
Guia Carmella B. Lara, Joseph Anthony C. Hermocilla, and Danilo J. Mercado. 2017. Exploring RSSI for UAV-to-GCS distance estimation. In *Proceedings of National Conference on Information Technology Education, Tacloban, Leyte, Philippines, October 2017 (NCITE 2017),* 6 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Unmanned aerial vehicles (UAVs) are powered devices that fly without a human pilot on board. They can be remotely controlled or autonomously flown using autopilot software. UAVs are popularly known as *drones*. They can carry loads which can be lethal or nonlethal[15]. UAVs were originally developed for military purposes and were used as weapons or surveillance equipment[9]. Recently, UAVs are being used in other civilian and commercial applications, such as, film making, construction, logistics, rescue, disaster management, and agriculture[17].

Two popular types of UAVs used for various commercial applications are the fixed-wing and multi-rotor. Fixed-wing UAVs are similar to airplanes in design while multi-rotor UAVs are similar to helicopters. Multi-rotors, specifically quadcopters, are becoming popular lately with several commercial vendors available on the market.[1][2][3]

When operated in remote mode, UAVs are manually controlled by a human pilot using a radio controller. In autonomous mode, UAVs depend on autopilot software that communicates with a Ground Control Station (GCS) using some protocol such as MavLink[4]. The GCS is the software installed on a laptop (ex. APM Planner[5]) to give the instructions for the UAV to follow, such as flying over a predefined flight path. UAVs rely on its sensors (barometer, accelerometer, gyroscope, compass) to be able to orient and position itself on its environment.

UAVs are also equipped with receivers that process radio signals from the Global Positioning System (GPS) satellites. The information from the satellites enable a UAV to determine its location by using information from at least four satellites[6]. However, the GPS receiver needs a clear view of the sky to work effectively. This makes GPS suitable for outdoor environments. GPS is accurate within +/- five meters, depending on the signal quality.

For UAVs that support IEEE 802.11 wireless networks(WiFi), distances can also be approximated using the Received Signal Strength Indicator (RSSI). RSSI also has its limitations when used in areas with walls and radio interference.

UAVs can operate in different flight modes. The Follow Me[6] mode is one example of autonomous flight mode that allows the UAV to follow the GCS that it is connected to using GPS information. The movement of the UAV is based on its relative position to the GCS. Given the altitude and ground distance, this flight mode controls the UAV to follow the GCS. In this mode, the GCS and the UAV must be both GPS-enabled. The difference in the GPS readings between the GCS and the UAV is used to estimate the distance between them. Follow-Me mode is widely used in cinematography to position the UAV relative to the subject.

This study investigates the combination of GPS and RSSI to provide a more accurate measure of distance between a UAV and its GCS in the Follow Me autonomous flight mode. A good combination will allow the UAV to fly more reliably in situations when one sensor fails or in more constrained environments with tighter spaces or partial view of the sky.

NCITE 2017, October 2017, Tacloban, Leyte, Philippines

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

[1] http://www.dji.com/
[2] https://www.parrot.com/global/drones
[3] https://3dr.com/
[4] http://qgroundcontrol.org/mavlink/start
[5] http://ardupilot.org/planner2/
[6] http://ardupilot.org/copter/docs/ac2_followme.html

BERTUD: BUILDING FOOTPRINT EXTRACTION AND REGULARIZATION IN LIDAR DATASETS THROUGH UTILIZATION OF A DISTRIBUTED SYSTEM

Ivan Marc H. Escamos[1], Allen Roy C. Roberto[1], Joseph Anthony C. Hermocilla[1], Miyah D. Queliste[1], Edwin R. Abucay[1], Gillian Katherine L. Inciong[1]

[1]University of the Philippines Los Baños, Los Baños, Laguna, Philippines
Email: ivanhescamos@gmail.com

KEY WORDS: Distributed Computing, Multi-core Processing, Disaster Risk Management

ABSTRACT: The Philippines is situated inside the pacific ring of fire, which makes it vulnerable to natural disasters such as typhoons, tsunamis, volcanic eruptions and earthquakes. Disaster risk management plans must be developed and updated to keep up with the effects of climate change. Currently, such plans in the Philippines often lack updated data on the location of infrastructure and residential areas. Thus, building footprint extraction is an important task.

One of the tasks of our project, UPLB Phil-LiDAR 1, is to extract building footprints in the Laguna and MIMAROPA areas of the Philippines. Manual digitization of building footprints using GIS is tedious. With limited personnel and an extensive area to cover, devising an automated workflow is important.

Through the use of remote sensing techniques, building footprint extraction can be automated. Light Detection and Ranging (LiDAR) is one of the most powerful remote sensing technologies nowadays. We have developed an automated workflow for building footprint extraction and regularization from LiDAR datasets. One of the problems encountered with the created workflow is its running time. A 1km X 1km LiDAR tile of an urban area would take hours to finish.

In order to address this problem, we developed BERTUD, a distributed system to enable full utilization of the available computing resources of our project. Distributed systems are systems that make networked computers finish a big task.

The system was able to fully utilize the computing resources of our project by providing two layers of maximization: macro-level and micro-level. On the macro-level, BERTUD divides large areas for building extraction to multiple computers effectively speeding up the process. On the micro-level, the slave program maximizes the available resources of its host computer by utilizing multiple CPU cores, without interrupting its current user.

1. INTRODUCTION

The Philippines is located inside the Pacific Ring of Fire, which makes it vulnerable to natural disasters such as typhoons, tsunamis, volcanic eruptions, and earthquakes. Every year, an average of 20 typhoons enters the Philippine area of responsibility. In 2013, the country was ravaged by one of the strongest typhoons in history: Haiyan (locally known as Yolanda). Around 16 million people were affected and about 12.6 Billion USD worth of damage was caused by Yolanda (GFDRR, 2015).

The effects of climate change clearly cannot be ignored. Thus, disaster risk management plans must be kept up to date. In the Philippine context, disaster risk management plans are rife with the lack of updated spatial data on infrastructure and residential areas. In the wake of advancing technology, methods of hazard exposure and disaster mitigation are continuously developing. Technologies such as Geographic Information Systems (GIS) and Remote Sensing (RS) are useful for this endeavor. One problem in hazard exposure is to identify man-made structures susceptible to hazards. Thus, building footprint extraction is an important task.

UPLB Phil-LIDAR 1 (http://phil-lidar.uplb.edu.ph) is a product of the concluded UP Disaster Risk and Exposure Assessment for Mitigation (DREAM) Program (http://dream.upd.edu.ph). Its main objective is to develop 3D hazard maps for the Philippine river systems in the Laguna and MIMAROPA areas of the Philippines. Another objective of our project is to extract building footprints in the said areas. Building footprint extraction is usually done through manual

OSv-MPI: A prototype MPI implementation for the OSv cloud operating system*

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

Eliezer A. Albacea
Institute of Computer Science
College of Arts and Sciences
University of the Philippines Los Baños
eaalbacea@up.edu.ph

ABSTRACT

In this paper we present OSv-MPI, a prototype MPI implementation to enable HPC applications that use the MPI standard to run on virtual machines with OSv, a new cloud operating system, as guest. OSv-MPI provides a library that can be linked to existing MPI applications and a set of utilities to execute the resulting binaries in an OSv instance. We tested our implementation using simple applications that use the primitives MPI_Send() and MPI_Recv() and show that correct results are obtained. We also present CPU usage statistics while running some of the applications.

CCS Concepts

•Networks → Cloud computing; •Computer systems organization → Cloud computing; Client-server architectures; •Software and its engineering → Message passing; Massively parallel systems;

1. INTRODUCTION

Computational scientists usually run their high-performance computing (HPC) applications on dedicated supercomputers or physical clusters. To speed up computations, parallel processing is used. A parallel programming environment is composed of a *job scheduler*, *process manager*, and *parallel library*. The job scheduler describes which resources(compute nodes) a parallel job, which consists of multiple processes, will run. The process manager starts and ends processes associated with the parallel job. The parallel library provides processes a mechanism to communicate[2]. Several parallel programming environments exist but the most popular and widely used is the Message Passing Interface (MPI) standard[4][1]. Open source implementations of MPI exists such as MPICH[2] and OpenMPI[3]. These implementations are available for various hardware architectures and operating systems.

Physical clusters however are expensive to procure, setup, and maintain. They are composed of physical machines (with their own power, CPU, main memory, and secondary storage) that are connected via high-speed networks such as Gigabit Ethernet or Infiniband. Cloud computing, *Infrastructure as a Service (IaaS)* in particular, provides an alternative such that setup and maintenance costs are reduced because of on-demand provisioning of virtual machines(VM)[8][1]. Cloud providers allow users to start and terminate VMs that run general-purpose desktop or server operating systems as guest. VMs are booted with disk images specific to a virtualization product such as KVM[4], Xen[5], or VirtualBox[6]. Figure 1 shows the list of disk images supported by the P2C cloud[6].

A running VM in a cloud is often referred to as *instance*. Virtual clusters are the equivalent of physical clusters in the the cloud. Amazon EC2[7], Microsoft Azure[8], and Google CE[9] are popular commercial or public cloud providers which provide customers access to VMs and clusters for a fee. Private clouds, which are used internally in an organization, are also common in research and academic institutions. These private clouds are deployed using open source cloud frameworks such as OpenStack[10](Figure 2). Instances in the cloud can run HPC applications as long as the guest operating system supports an MPI implementation. Figure 3 shows an example MPI application running on a virtual cluster using MPICH[6].

The adoption of the cloud for running HPC applications however is still limited[9]. Since virtualization adds another layer (the hypervisor) above the physical hardware, some performance degradation is inevitable, particularly in the networking stack. In addition, general-purpose operating systems that run on the virtual machines add more abstraction layers (process management, file system, networking, etc), further degrading performance. To address this, several approaches have been proposed in the literature and

*Code:http://srg.ics.uplb.edu.ph/resources/downloads/osv-mpi-source-ncite2016.zip

[1]https://www.mpi-forum.org/docs

[2]https://www.mpich.org
[3]https://www.open-mpi.org
[4]http://www.linux-kvm.org
[5]https://www.xenproject.org
[6]https://www.virtualbox.org
[7]https://aws.amazon.com/ec2
[8]https://azure.microsoft.com
[9]https://cloud.google.com/compute
[10]https://www.openstack.org

NCITE 2016 Dipolog City, Zamboanga del Norte, Philippines

Sustainable Application of Versatile Electronic System of Billing of Electricity and other Utilities Software and Technology

Ivy Joy U. Aguila
Institute of Computer Science
College of Arts and Sciences
University of the Philippines
Los Baños
iuaguila@up.edu.ph

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

Susan G. Tolentino
Utilities Billing Section
RGDO-OVCPD
University of the Philippines
Los Baños
sgtolentino@up.edu.ph

ABSTRACT
Billing and collection of fees are essential activities in an organization that provides services because they directly affect the income generated and doing the processes manually is more prone to errors particularly in encoding data. We present in this paper the design and implementation of SAVES-BEST, a web application that automates the billing and collection activities for utilities, such as electricity and water, provided by an academic institution to its constituents. In this system, after a single data entry, bill and receipt preparation transactions can be made easily, and reports are generated instantly. Feature and transaction processing evaluations of SAVES-BEST show that stakeholders are more likely to use it compared to the system currently being used.

General Terms
Information systems∼Enterprise applications,
Applied computing∼Business process management systems,
Social and professional topics∼Automation

Keywords
utilities billing, collection monitoring, process automation, web application

1. INTRODUCTION
Since the time that computers are used in offices, useful programs for staff have been created to lessen their manual labor. Vouchers are typed on a text editor, printed out with a printer and then passed to be signed and approved. In accounting, ledgers are typed on a spreadsheet editor which automatically computes with the given formula. But this scenario is from 40 years ago. In addition to that, a survey result [17] shows that almost one working week was lost per employee just because they were finding lost hard copy documents. There is also a bigger chance of having errors and inconsistencies on data since it is to be copied manually from one office section to another.

Nowadays, the trend is to make every office paperless, to automate processes as much as possible, and to help ease decision making of the management. Many solutions company offer this kind of all-around business process automation system already. But because of the costly amount of these products and associated maintenance costs, the university decided to create its in-house system for automating billing and collection activities.

This paper describes the design, implementation, and evaluation of a billing and collection system that will be used by the UPLB Utilities Billing Section. The in-house system will not just automate the office processes but also save the university thousands of pesos in costs from purchasing existing solutions. The monitoring and reporting modules included in the system can also be used to alert the administration so that problems can be resolved immediately which will also help the university save some resources.

2. BACKGROUND
The Utilities Billing Section is in charge of billing electric, water, and garbage of more than 340 clients subscribing to UPLB utilities services and are also in charge of monitoring payments of such services. The responsibilities of this office section are very crucial because the university's monthly collection is more than a million pesos, which is a major source of income for the university.

The stakeholders are composed of Utilities Billing Section staff and the consumers of utilities under the university. The Utilities Billing Section has a personnel for preparing the bills for each month and the reports related to the billing process. There is also a personnel who takes care of making receipts and monitoring consumer payments. As part of the staff, three meter readers collect meter readings every 22nd day of the month along their assigned routes. They also deliver the printed bills at a later date. The consumers are grouped according to price per unit of consumption: Res-

Is SystemOne "Slow"?

Rommel V. Bulalacao , Ivy Joy U. Aguila, Joseph Anthony C. Hermocilla
Institute of Computer Science
rvbulalacao@up.edu.ph, iuaguila@up.edu.ph, jchermocilla@up.edu.ph

SystemOne Decaf (S1) is a web application running on Linux-Apache-MySQL-PHP(LAMP) stack that supports the student registration process of UPLB. It is accessed over the Internetand deployed using the Client-Server application architecture. In this paper, we will answer the frequently asked question by its main users, the students: Why is SystemOne "slow"?

It is easy to say that a system is slow. But it will be more convincing if it said with some quantitative proof. Thus, in this work, we present some numbers that can quantifywhetherS1is indeed "slow", or not.

We are interested in measuring the Quality of Service (QoS) of S1 which is defined in terms of response time, throughput, and availability. This is done by loadtesting S1, sending requests that emulates the behavior of the students during the peak of the registration period to the server via scripts, and obtaining measurements. An overview of S1's software architecture is presented to enumerate the different paths taken by clients' requests to the server, and back. These paths affect the QoS. We obtain measurements for different system deployment configurations for comparison. The result of load testing can tell us the best system configuration for S1 and more importantly it can tell us why it is slow or if it is indeed slow. It can also give us some insights on how and where we can improve the system.

Initial results of our laboratory tests indicate that the main bottleneck is in the storage/database(MySQL) component of S1. This result is validated by analysing logs obtained from actual deployments of S1 for the past semesters.

Keywords: web application architecture, workload, quality of service, response time, throughput

COMPARISON OF DIFFERENT MACHINE LEARNING CLASSIFIERS FOR BUILDING EXTRACTION IN LIDAR-DERIVED DATASETS

Ivan Marc H. Escamos[1], Allen Roy C. Roberto[1], Edwin R. Abucay[1], Gillian Katherine L. Inciong[1] ,Miyah D. Queliste[1], Joseph Anthony C. Hermocilla[1]

[1] University of the Philippines Los Baños, Los Baños, Laguna, 4031, Philippines
Email: ivanhescamos@gmail.com

KEY WORDS: Object Based Image Analysis, Feature Extraction

ABSTRACT: Building extraction in remotely sensed imagery is an important problem that needs solving. It can be used to aid in urban planning, hazard assessments and disaster risk management among others. Light Detection and Ranging or LiDAR, is one of the most powerful remote sensing technologies nowadays. Many studies have used the fusion of LiDAR data and multispectral images in detecting buildings. This study seeks to maximize the power of LiDAR imagery to be able to classify buildings without the aid of multispectral imagery. This work follows the Object Based Image Analysis (OBIA) approach. Instead of the traditional pixel-based classification methods, pixels are segmented into logical groups called objects. From these objects, features for building extraction are calculated. These features are: the number of returns, difference of returns, and the mean and standard deviation of positive surface openness. These objects are then classified using different machine learning classifiers such as Support Vector Machines, K-Nearest Neighbors, Naïve Bayes Classifier, Decision Trees, and Random Forests. A comparative assessment was done on the performance of these different machine learning classifiers. The classifiers performed similarly with the Random Forest Classifier slightly outperforming the others.

1. INTRODUCTION

Building extraction in remotely sensed imagery is an important problem that needs solving. It can be used to aid in urban planning, hazard assessments and disaster risk management among others. The development of a building extraction method is one of the objectives of the UPLB Phil-LiDAR 1 project.

Light Detection and Ranging (LiDAR) is one of the most powerful remote sensing technologies nowadays. The LiDAR is an instrument similar to radar, but it uses laser pulses instead of radio waves. (Campbell, 2011) Geospatial products, like Digital Elevation Models (DEMs), can be created using LiDAR instruments on aircraft (NOAA, 2015).

Many recent studies have used LiDAR in conjunction with multispectral imagery. The different frequencies of the electromagnetic spectrum provide a lot of information helpful for building extraction and remote sensing in general. But high resolution multispectral imagery is not easily accessible. In UPLB Phil-LiDAR 1's case, orthophotos corresponding to LiDAR DEMs are available. These orthophotos have issues like misalignment with the LiDAR DEMs, areas with no coverage, inconsistency of colors due to changing atmospheric conditions, etc.

Our study follows the Object Based Image Analysis (OBIA) approach. Instead of the traditional pixel-based classification methods, pixels are segmented into logical groups called objects. (Blaschke, 2010). From these objects, features for building extraction are calculated. Features are quantities that describe an object. (Kohavi, 1998.) Machine learning classifiers, together with these features are used to classify objects.

The general objective of this study is to evaluate different machine learning methods in building detection by object based image analysis using only LiDAR derivatives. The specific objectives are as follows:

a. To identify features from LiDAR derivatives suitable for building extraction;
b. To extract buildings using different machine learning classifiers; and
c. To evaluate and compare the different machine learning classifiers

2. REVIEW OF RELATED LITERATURE

Through the years, airborne LiDAR technology has reached new heights of interest in remote sensing especially in applications such as building extraction. A number of techniques have been developed to correctly identify buildings, and these methods vary in performance and precision.

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

P2C: Towards Scientific Computing on Private Clouds

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

ABSTRACT

Peak-Two Cloud (P2C) is an Openstack-based private cloud on top of commodity hardware targeted for scientific and high-performance computing. We describe how it was configured and deployed then present use cases for education and actual production runs. We also introduce our own tools, vcluster and vhadoop, for rapid and on-demand deployment of MPI and Hadoop clusters, respectively, on P2C. Finally, we report some benchmarks results on clusters deployed using these tools.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems—*Network Operating Systems*

General Terms

Management, Measurement, Performance

Keywords

Cloud Computing, High-performance Computing, Clusters

1. INTRODUCTION

Cloud computing has become a buzzword in today's modern computing although there is no agreed upon meaning of the term. NIST published a definition that is widely quoted and used[19]. The popularity of cloud computing mainly comes from its ability to provision additional resources on demand with minimal intervention from the provider. It leverages advances in virtualization and web service technologies. For example, an owner who observes a sudden increase in workload on his website can start another virtual server machine (called an instance) almost instantaneously to accommodate the additional load. Extensive discussions of what cloud computing is can be found in the literature[12][20] [11][24].

A cloud can be deployed in several ways, depending on who can access the services it provides. Private clouds are operated for an organization. Community clouds are shared by several organizations to support a community with shared concerns. Public clouds are available to the public. Lastly, hybrid clouds are composition of two or more clouds [19].

Cloud computing offers service models which include Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), and Infrastructure-as-a-Service (IaaS). Most are familiar with SaaS as it provides user functionality directly, Google Docs and Dropbox being notable examples. Developers on the other hand will be more acquainted with PaaS because they use APIs to develop applications in which Google App Engine is an example. IaaS allows the consumer to provision computing resources (hardware,network, storage) to run arbitrary software including operating systems[19]. A popular IaaS public cloud is Amazon's Elastic Compute Cloud (EC2) and Simple Storage Service (S3). Most IaaS providers use proprietary technologies in their implementation. Lately, a number of open source frameworks have been released for deploying IaaS private clouds. This work focuses on IaaS private clouds using Openstack described in the next section.

Clouds today are used mainly for hosting web sites and deploying online services (web applications). They provide instances with operating systems that can run web server software, scripting engine, and database management systems. Linux-Apache-MySQL-PHP (LAMP) stack is a typical configuration for a cloud instance.

The success of this technology is enormous and people are still looking for uses beyond web application hosting. The science community is one group that is interested in leveraging the use of cloud. They are looking at the possibility of running entire scientific applications on the cloud. However, these applications are compute and communication intensive compared to web applications and are often run on distributed systems and clusters. The setup of these requires highly specialized skills which a typical scientist will not have.

Thus, the main goal of P2C is to provide a cloud environment usable for scientific applications. It aims to harness commodity hardware used in teaching laboratories to maximize their utilization since most of the time they are used for simple programming purposes only. This approach of using teaching laboratory machines eliminates the upfront cost of purchasing dedicated hardware for clusters, which also saves on power consumption. In addition, dedicated clusters are often idle since they are not used as frequent as

Squidler: A Proxy Log Parser with Automatic Web Page Classifier

[1]**BARON GERARLD S. MANZANO,** [2]**Ludwig Johann B. Tirazona,** [3]**Joseph Anthony C. Hermocilla**
Institute of Computer Science, UPLB
Information Technology Center, UPLB
[1]bgsmanzano@gmail.com,
[2]ljtirazona@gmail.com,
[3]jachermocilla@gmail.com

Internet connectivity is an expensive but very important resource especially in academic institutions. Its non-academic use deprives others of the bandwidth needed to conduct research and other academic-related tasks. Thus, there is a need to determine and classify the web pages visited by constituents in order to properly manage access to this costly resource. Squidler is a web application that combines the functionality of a web proxy log parser and a web page classifier. It automatically retrieves the access log from a proxy server then outputs a web page that displays information about outgoing HTTP requests from clients. It then downloads the web pages from the URLs in the log, simulating browser requests, and then performs the classification. Our method uses Support Vector Machines (SVM) to automatically classify web pages either as academic or leisure. We tested the application using the proxy server deployed in the University of the Philippine Los Banos. Results showed that Squidler was able to classify most of the academic pages but failed on some leisure pages. Also, some performance issues were encountered in relation to the amount of data that can be processed simultaneously.

1. Introduction

The Internet has become the main source of information for our everyday lives. For businesses, communications, and academics, a lot of information is stored and retrieved in the Internet. People gain access to the Internet by connecting to private connections or public servers using a Web browser. These servers are hosted by organizations that provide services and manage resources for the benefit of all. These servers, or proxy servers, act as gateways between a private network and the Internet. When a computer, or client, connected to the server tries to access a website, the request is sent to the proxy server, and the proxy server sends the request to the actual server hosting the website. After receiving the request, the server sends the information needed back to the proxy server, and then back to the client who requested it. The proxy server also saves a cache, or a copy, of the data requested, for faster access later [1]. Proxy servers can also keep track of cache refreshing, garbage collection for old cache files, as well as selecting specific documents to be cached [2]. The Information Technology Center (ITC) in UPLB uses a proxy server as well. It uses Squid, a widely used proxy server for Linux and UNIX platforms[1]. They also use Squint, a Squid proxy log parser that checks their network traffic, and it is viewed in HTML. They can view who are the ones connected to the server, and for how long they are connected. It also displays the amount of bandwidth and resources they have consumed. It can also display the different websites visited.

The resources provided by proxy servers are shared to everybody accessing the server, so when a client connected to the server requests for websites with high bandwidth and the resources cannot keep up, the connection slows down and everyone in the server is affected as well. Extreme cases of resource hogging could make the server reject all service requests, which would make matters worse for the clients and the administrators. Prohibiting all high-bandwidth sites is a solution, but it may not be much helpful when the sites in question are needed for academic purposes. Not knowing what kinds of sites to allow or forbid can lead to problems later on.

A useful extension to log parsers is a web page classifier, an application that sorts web pages according to different categories. The administrator can sort the different websites that the clients visit, as well as pinpoint the categories that consume the most bandwidth. He can potentially control what kinds of websites should be allowed or forbidden. This paper focuses on integrating and demonstrating these two applications together: a squid log parser and a web page classifier into a web-based application. The output of the application would display information about the clients connected to the server, as well as the websites visited and the categories they fall under.

The main objective of this work is to combine the functionalities of a Squid log parser and a web page classifier in one web-based application. The specific objectives are:

1. to create a web-based squid log parser that displays information of the IP addresses and their connections;
2. to gather data from the websites visited by the IP addresses and classify them according to subjects (business, leisure, etc.); and
3. to test the final application in a working environment.

2. Review of Related Literature

Research and development of proxy servers is still ongoing. These proxy servers usually have logs stored in them that contain information. It is up to a separate log parser application to make use of this information.

Automatic Network Bandwidth Control Application for Linux-based Internet Connections

Korinto Miguel G. Aguinaldo and Joseph Anthony C. Hermocilla.

I. Introduction

A. Significance of the Study

With the advent of data sharing and synchronization over the internet, many applications have become dependent on bandwidth allocation in order to function properly. Much of current user-friendly software requires frequent to constant internet connection to fully realize their primary function. The usual examples of this are real-time network games such as World of Warcraft and BitTorrent services. These kinds of programs may be given a larger bandwidth allocation, depending on their setting. In a network of computers using a common port gateway, a larger bandwidth allocation for one computer usually generates smaller bandwidth allocation for others.

Bandwidth control or traffic shaping is a necessity in maintaining a fair bit-rate for all computers in the network. Through bandwidth control, the bit-rate of a computer can be throttled down in order to give larger bandwidth allocations for other computers. The most important task in network administration is to regulate the connection of the network to internet services and to every connected computer. It is the prerogative of the administrator to be able to know how he or she will achieve a balanced bit-rate for all of the computers and when a computers connection should be sped up or throttled down.

Recent Linux distributions have the shell command *tc* which allows the user to filter and throttle network traffic. However, this tool may present confusion to those who are new to networking concepts because its command syntax may be complicated and hard to understand.

The purpose of this project is to produce a user-friendly application that serves as an command generatorfor *tc*. The application must present *tc* syntax in a way that is easily understood, and can be modified by keyboard and mouse. Finally, the application must be able to automatically apply *tc* commands so that the network administrator will not have to constantly monitor the network so that fair bit-rate for all computers is achieved.

B. Statement of the Problem

Network administration involves regulation of internet use. There are tools available to do this, but these tools are either difficult to be used or too complex to be understood by inexperienced administrators. A simple program that could integrate known tools such that it enables the administrator to easily maintain fair use of the internet connection is ideal.

Presented to the Faculty of the Institute of Computer Science, University of the Philippines Los Baños in partial fulfillment of the requirements for the Degree of Bachelor of Science in Computer Science

C. Objectives of the Study

The general objective of the Special Problem is to alleviate the difficulties involving network administration.

Specifically, the aim is:

1) To develop an application that can automate traffic control; and
2) To develop a user-friendly graphical user interface for the application.

II. Review of Literature

Linux-based operating systems is popular in computer networking, especially when it is known that the Linux networking attributes are configured and inspected completely. [1] We can infer that popular, even local, networking services are usually run from Linux servers, especially when the functionalities and instructions for handling these programs are explicitly included in their installation. This allows scrutiny for young network administrators.

The other computers will be dependent on the server for their internet connection. The server, or router terminal, may allocate bandwidth to itself and other computers, or ”dependent terminals”, in the network.

Imagine that these dependent terminals are being used by members of an organization, as employees or students. Fisher implies that most of these kinds of people usually visit social networking sites such as Facebook and Twitter first before proceeding to the actual purpose of their internet access. [2] Also, commonly during their breaks, these people access sites involving streaming media such as entertainment and news sites, which are usually encountered in the internet. [3]

According to a statistic shown by a White Paper from Symantec MessageLabs, sites that has streaming media content is one of the highly preferred to be blocked by administrators, along with advertisement and popup-generating sites. [4] The common awareness of why this happens is because these sites take up a lot of bandwidth. Streaming media and other kinds of download services make up an increasing percentage of traffic in the internet. [5] Such high-bandwidth connections take a toll in local networks, especially when many dependent terminals access these sites at the same time such as during breaks.

SquiGIS: A web-based Geographic Information System for the UPLB Network

Shiela Kathleen L. Borja , Ludwig Johann B. Tirazona , and Joseph Anthony C. Hermocilla

***Abstract*— The researchers developed a system which resides on a machine different from the one keeping the Squid access logs. Both machines belong to the same network. This was done to allow larger memory allocation for the system. The researchers created a log parser to get the logs from the machine containing the Squid access logs through SSH every two hours. The parsed data is stored in a database managed by squiGIS. The data stored in the database was used to analyze the logs and create a web-based monitoring module for the system. Search functionalities specified by the UPLB Information Technology Center were also incorporated into the system.**

***Index Terms*— Geographic Information System, network monitoring, squid log parser and analyzer**

I. Introduction

A. Background of the Study

The University of the Philippines Los Baños(UPLB) network has a very essential role in rendering the university's services to the students, faculties, and staffs effectively. Thus, it is very important to maintain the network's integrity and performance through an effective monitoring system. Logs are files maintained by the proxy servers to keep track of the data about the users and their Internet activities. Data from the logs are extracted and used as input to the monitoring system to generate reports about the Internet activities of the users.

Currently, the UPLB Information and Technology Center (ITC) uses Squid 3.0 as the proxy server for the UPLB network. A proxy sever has two main functionalities in the network. The proxy server acts as an intermediary between the client and the server containing the requested resources and requests for the resources in behalf of the client. It can also be used to restrict access to undesired sites.

The objective of this project was to build a monitoring system for the Internet usage in the UPLB network through processing the logs generated by the Squid proxy server used in the UPLB network. Then, the GIS was used to help monitor the Internet activities in the network at a certain time.

B. Statement of the Problem

ITC uses Squint, Calamaris, and LightSquid as log parsers but no single log parser can suffice the requirements that ITC needs to monitor the activities in the UPLB network.

The Squint log parser is an open-source Squid log file analyzer which produces static HTML reports. [1] Calamaris is a log parser for Squid, NetCache, Inktomi Traffic Server, and other proxy servers and generates reports including Peak-usage, Request Methods, Status Report of incoming and outgoing requests, second and Top-level destinations, content-types and performance. [2] [3] LightSquid is a perl-based cgi Squid logfile parser. [1]

ITC uses the all of these three log parsers in order to combine the good features of each log parser and analyze the Internet activities in the network. Squint is used more often because it is faster than LightSquid but Squint is also inefficient because it eats up a lot of CPU time [4]. Calamaris analyzes the overall statistics of the logs but it does not analyze individual information of each client.

Although Squint, Calamaris, and LightSquid are powerful tools, UPLB ITC needed a more customized log file analyzer to better suit the unit's monitoring transactions.

C. Significance of the Study

This study aimed to help improve monitoring of the Internet resources through adding the necessary functionalities as required by the UPLB ITC. Improved monitoring of the Internet activities in the network would help track violators. Logs and reports about the internet activities of the violator could serve as a hard evidence against the violator and could also serve as a basis for the proper sanctions that would be given to the violator. [5]

This study also aimed to help ITC to use only one log parser which could cater all their requirements since the specifications of the system depend on their requirements. Additional features such as the list of *View by Top Users According to bandwidth consumed, Top IP Addresses According to bandwidth consumed, Most Accessed Domain, View by IP Address, View by Lightweight Directory Access Protocol (LDAP) User name, View by Subnetwork, View by Active Hosts, Top IP Addresses with most Access Denies* , and *Show by Date* were included to improve the searching functionality of the system. The locator could also help in visualizing the Internet activities in the network since the data about the number of requests and total bandwidth for each *Active Host* were plotted on the UPLB map. Using the locator, the administrator could have a quick look on the distribution of the client requests during a certain time of the day.

D. Scope and Delimitation

The system was limited to mapping wired connections, that is excluding clients connected through wireless connection

Presented to the Faculty of the Institute of Computer Science, University of the Philippines Los Baños in partial fulfillment of the requirements for the Degree of Bachelor of Science in Computer Science

Eucalyptus LSS: Load-Based Scheduling on Virtual Servers Using Eucalyptus Private Cloud

Shenlene A. Cabigting and Joseph Anthony C. Hermocilla

***Abstract*— The University of the Philippines Los Baños currently offers a number of applications to its students and staffs. However, a limited number of servers and hundreds of users accessing different services at the same time would results to heavy traffic and a huge response time. A private cloud was setup to enable easy and flexible deployment of virtual servers within the network using Eucalyptus Private Cloud. A load balancer was deployed in the cloud to properly distribute task among the virtual servers, provide faster response time and prevent the possibility of data lost due to server failure. The load balancer was developed using Java and is composed of four classes: MyServlet, Balancer, Server and Request. The load balancing scheme used took into consideration not only the load and capacity of the servers but also the resources needed by each incoming request. A graphical web interface was also provided for the network administrator to visualize and monitor the current status of each virtual machine in the cloud.**

***Index Terms*— Eucalyptus, cloud computing, load balancing, virtualization, server scheduling**

I. Introduction

A. Background of the Study

Cloud computing, while still at its early stage, is already making a scene in the Information Technology (IT) industry. Continuous development in high-speed broadband internet enables this new technology to offer possibilities to both its end-users and service providers.

Cloud computing is the means by which services can easily and conveniently be accessed from a shared pool of configurable computing resources through the internet or a local area network [1]. These resources make up what is known as the cloud and consist of both the hardware and the software in the datacenters from which services are provided [2]. A cloud user must send a request to the cloud to access these services. Whenever such request is granted, a fraction of the cloud is provisioned to the requesting user and will remain dedicated to that user until it is released [3].

One way of classifying clouds is based on the portion of the resources delivered as a service. They are referred to as the three service models namely: Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). They are also sometimes referred to as the cloud layers where IaaS is the lowermost layer.

SaaS provides the user access to applications deployed in the providers cloud infrastructure through a client interface and is often referred to as Software on Demand. In this service model, the user has no control over the underlying cloud infrastructure. PaaS provides the user the ability to create, manage and/or control applications in the cloud. These applications can either be created by the user or acquired from other services as long as they are supported in the cloud environment of the service provider. IaaS provides the user with their own virtual cluster on which they can provision processing, storage, networks and other fundamental computing resources and on which they can deploy and run applications and even operating systems. [1]

Another way of classifying clouds is based on the nature of access and control with respect to how the provisioned resources will be used [3]. This classification is more commonly known as the deployment models and includes the following: private, community, public and hybrid. Private cloud refers to the internal datacenter of a company or organizations sharing the same concerns. These organizations may have the same policy, security requirements, compliance considerations, etc. The public cloud infrastructure, unlike the first two deployment models mentioned, is made available to the public and is usually owned by companies or organizations selling cloud services. A hybrid cloud is the resulting infrastructure when two or more of the other deployment models (private, community and public) are combined. In this deployment model, each cloud component remains unique yet they are bound together by a standardized technology. [1]

One key characteristic of cloud computing is scalability. A cloud must be able to accommodate rapid increase or decrease in demands and elastically scale in or scale out computing resources based on these demands [3]. Applications deployed on a cloud should be available to any number of users at any time. To achieve this goal, virtualization and load balancing must be supported in every cloud.

Virtualization is the ability to run instances of virtual machines (VM) on top of a hypervisor. A hypervisor provides a uniform abstraction of the underlying physical machine allowing multiple VMs to execute simultaneously. [3] Virtualization provides the cloud the flexibility to allocate and deallocate computing resources to satisfy user requests. Load balancing, on the other hand, is the process of transparently and efficiently distributing user requests to the number of available servers [4]. With these two working hand in hand, cloud computing creates the illusion of an infinite computing resources available on demand to the users at any time [2].

Presented to the Faculty of the Institute of Computer Science, University of the Philippines Los Baños in partial fulfillment of the requirements for the Degree of Bachelor of Science in Computer Science

A Network Intrusion Testbed through Honeypots

Luisse Margarette A. Macasaet and Joseph Anthony C. Hermocilla

***Abstract*— The field of honeypots is fast evolving and researchers are trying to find more innovations for this technology due to its behavioral analysis capabilities of network intrusions which complements the traditional signature-based detection methods. This paper presents the effectiveness of Honeyd, a low-interaction honeypot, when used as a deceptive tool to lure attackers into thinking that they have found a vulnerable segment in the network and their actions are far from being monitored.**

I. Introduction

Malicious hacking has been a problem since the time when wardialing and phone phreaking were all the hype. Back then, very little was known about these intruders, let alone tools to detect and prevent their attacks. But at this present age of script kiddies and blackhats, we heavily rely on access control tools and other security policies to prevent unwanted access to our private data [1].

A. Background of the Study

Network intrusion detection is concerned with the detection of attacks made against a network that are meant to compromise and exploit the confidentiality, integrity and availability of a resource [2]. The main concern of network intrusion detection, however, is to identify malicious network activities and differentiate them from normal network activities.

Throughout the years, many technologies and tools have been used to create and test systems that perform automated intrusion detection, or simply intrusion detection systems (IDS). One of these is the honeypot. A honeypot is a trap which usually consists of data, a computer or a network site. It appears to be a normal part of a network or more often than not, an important part of a network that contains data valuable to hackers; but in reality, it is actually isolated and monitored for malicious activity [3]. More precisely, a honeypot is an information system whose value lies in unauthorized or illicit use of that resource [4].

B. Statement of the Problem

Since hackers and virus writers have come up with better ways to evade anti-virus technology throughout the years, the use of signature-based anti-virus software is proving to be less effective in putting a stop to malicious codes running in our computers. There is a need to find a way to analyze malicious activity without having to rely on the traditional signature-based anti-virus tools but instead, complement what these tools can already do.

Presented to the Faculty of the Institute of Computer Science, University of the Philippines Los Baños in partial fulfillment of the requirements for the Degree of Bachelor of Science in Computer Science

C. Significance of the Study

It is necessary to use a honeypot instead of a firewall because the detection algorithm that Network Intrusion Detection Systems (NIDS) use is based on how signature-based anti-virus tools detect malicious activities. They both rely on a database of attacks that have already been detected and recorded. This leaves NIDS unaware of newly-developed compromises that are unknown to it at the time of the attack. A honeypot, however, can detect vulnerabilities that are not yet identified.

There is a need to create a testbed that will totally involve a network under study and at the same time, prevent intrusions from exposing and exploiting the vulnerabilities of the said network. Since honeypots are deceptive systems, it will be very useful in hiding the real value of the data that pass over the network.

This study will make use of a honeypot in the creation of a testbed that will help in testing and identifying vulnerabilities of a network. It may also help in the analysis of attacks that are both known and unknown to the public.

This will be different from previous of testbeds of the same nature in a way that the honeypot will not simply be a part of a network that is connected to the internet and will wait for attacks or malicious activity but it will also be attacked deliberately by the tester.

D. Objectives of the Study

The main aim of this study is to implement a network intrusion testbed that uses honeypots. Specifically, this study aims to:

1. Create an interface that will make the configuration of a honeypot easier;
2. Use a honeypot configuration that will be tested over a small network; and
3. Analyze the activity that will be logged by the honeypot

II. Review of Related Literature

In 2001, when Code Red [5] was detected on the internet, Liston had the idea of a sticky honeypot [6], [7]. Thus, the LaBrea Tarpit was born. The LaBrea Tarpit uses unused IP addresses and creates virtual servers on them. These virtual servers respond to connection attempts that are made by attackers. This action delays the attackers until they get stuck for a period of time. This is why it is called a tar pit.

Network-based detection refers to methods used to help detect malicious entities by studying network traffic [8]. Szor [9] proposed to update and maintain a list of hosts or network segments that are allowed to access the resources of a network. Through this, packets from intruders are simply not allowed

Parallel FlowViz

Angel Manica V. Raquel and Joseph Anthony C. Hermocilla

***Abstract*— This project improved FlowViz, a flood simulation software. It applied the concepts of parallel computation to adapt for wider area simulations. Using Data Elevation Models or DEMs as data input this program computes for the possible catchment areas using the Triangular Multiple Direction (MD∞) water delineation method and outputs a 3D visual representation of the waterflow. It was designed to use parallel computation where the master node was assigned for the division of data into subgrids which can then be processed by separate nodes locally and then sent back to the master node for the visual representation of the data.**

***Index Terms*— Parallel Computing, DEM, Triangular Multiple Flow Direction Algorithm**

I. Introduction

A. A. Background of the Study

The main aim of this project is to improve FlowViz , a flood simulation software [1] . The data used in FlowViz was limited to only the mountain areas of Luzon and this project will aim to use a larger scope of location in every place in the Philippines with available Digital Elevation Models(DEMs). Flood simulation involves a great deal of computation and with a larger data set, this project seeks to reduce the computation time by modifying FlowViz to apply parallel computation for a wider area simulation.

B. B. Statement of the Problem

A flood is a hydrological event characterized by high discharges and/or water levels that can lead to inundation of land adjacent to streams, rivers, lakes, wetlands and other water bodies. It is the costliest natural hazard in the world. It can cause damage to property and/or human lives. What is more, there has been an increasing trend associated to the number of deaths caused by flood [2]. Since floods are mostly caused by natural factors, the extent of their damages can be difficult to predict, but with relevant information, planning ahead can be done to minimize the damages they may bring.

C. C. Significance of the Study

Flood models are used to simulate how rain water flows overland to determine the probability of flooding [3]. They can also be used in planning and designing structures so that pre-emptive measures can be taken in time . Early flood warning systems serve to save lives, minimize flood damage to properties and reduce economic and social losses [4]. Awareness in the probability of flooding in an area should give time to construct temporary defences, to evacuate people if necessary and to move valuables from buildings and properties. Also, not all floods are harmful especially those that are small enough to just create wetlands which can be useful in agriculture. FlowViz is software developed to model the surface water flow over a wide area based on digital elevation models. Being able to model the surface water flow will allow the identification of possible catchment areas where flood might occur in the future. This study will aim to improve FlowViz by using a different method in catchment computation and by incorporating the concept of parallel computing.

Presented to the Faculty of the Institute of Computer Science, University of the Philippines Los Baños in partial fulfillment of the requirements for the Degree of Bachelor of Science in Computer Science

D. D. Objectives of the study

The main aim of this project is to improve FlowViz. Specifically, the study aims to:

- Use parallel computation in the simulation,
- Improve the Visualization of the Study Areas, and
- Implement the triangular multiple flow direction (MD∞) algorithm.

E. E. Date and Place of Study

This study was conducted in the Institute of Computer Science, College of Arts and Sciences, UPLB from November 2011 to March 2012. Using the MPC cluster, one of the Grid services provided by the Advanced Science and Technology Institute (ASTI) High-Performance Computing (HPC) Facility and Philippine e-Science Grid (PSciGrid).

II. Related Work

Through Digital Elevation Models (DEMs), hydrology determines the path of water/flow directions. One of the earliest methods in tracing water flow is the D-8 algorithm, which was introduced by OCallaghan and Mark (1984). This Method was implemented in FlowViz [1]. Several grid-based flow routing algorithms have already been presented by different authors after the development of D8 (Quinn et al. 1991, Freeman, 1991; Holmgren 1994; Mitasova and Hofierka, 1993; Mitasova et al. 1995, 1996; Tarboton, 1997). The triangular multiple direction (MD∞) algorithm presented by Seibert and McGlynn (2007) combines the benefits of triangular single direction flow (D∞) and the other multidirectional flow algorithms. It is more appropriate than the other existing flow algorithms across a range of landscapes, DEM resolutions and applications. FlowViz was originally implemented as a sequential program, and so as many other currently existing programs. Although the speed at which sequential computers operate has been improving at an exponential rate for many years, the improvement is now coming at greater and greater cost. As a consequence, researchers have sought more cost-effective improvements by building parallel computers which

Using Compute Unified Device Architecture (CUDA) in Parallelizing Different Digital Image Processing Techniques

Mikaela Nadinne A. Silapan and Joseph Anthony C. Hermocilla

***Abstract*— Graphics Processing Units (GPUs) have been conventionally used in the acceleration of 2D, 3D graphics and video rendering. Because of its performance and capability, the GPU has evolved into a highly parallel programmable processor that specializes in memory bandwith utilization and intensive computation. For operations involving graphics, GPUs offer a lot of gigaflops of processing prowess.**

The programmability and competency of GPU in the field of general-purpose computing is exemplified through the implementation of image processing techniques using Compute Unified Device Architecture (CUDA) which makes image processing be implemented with a high-speed performance.

I. Introduction

Different fields of scientific study use images to extract and emphasize significant data. An image is a function of two variables, $f(x, y)$, where x and y are spatial coordinates. The value of f at a coordinate pair (x, y) is called the gray value or intensity of the image at that point. Each pair of coordinates (x, y) is called a pixel. Images can contain hundreds of thousands of pixels. A digital image is created when the values of f for x, y and its intensity are finite [1]. The digital images may be modified, enhanced and processed for various purposes and with different techniques. These techniques are then implemented using some language and the data produced are used for analysis.

The area of study which refers to processing digital images using digital computers is known as digital image processing. One of the most important application of image processing is the enhancement of visual information. The implementation of digital image processing techniques has been successful through the use of computers. However, the computations required for processing digital images involve intensive geometrical and mathematical calculations. In addition, it will make the processor do these calculations for all of the pixels in the image; thus, while computing for values, the processor accesses the memory in a higher rate. It is an operation wherein large matrices are involved, assuming that the image is a two-dimensional array with each cell as a pixel.

The objectives of the field of improving visual perception, including digital image processing, using computers have long been achieved by man [1] through two-dimensional and three-dimensional graphics rendering, simulations, displays and etc. One of the most important tools that lead to the outstanding innovation in visual computing today is the graphics processing unit (GPU).

GPUs were designed for single instruction execution within multiple machines and for accessing multiple pixels in parallel to enhance the quality of computer graphics. Inspite of limitations in executing a single instruction, advances in the capability of GPUs has been already within our reach. As a result, GPU has become a tool not only for graphics rendering, giving rise to General Purpose GPU (GPGPU) computing [2].

A. Significance

During the early days of computers, computer graphics became important for the representation of results written by different kinds of programs. Even though the CPU can satisfy thedemands of some programs using only its resources, some programslook for highly accelerated 3D graphics pipeline [3].

In 2003, standard GPUs with 32 bit floating point numbers and programmable Vertex and Fragment processors were introduced in the market [3]. According to Owens et al., the modern GPU is not only a powerfulgraphics engine but also a highly parallel programmable processor featuring peak arithmetic and memory bandwidth [4]. The GPU is a processing unit built to accelerate the rendering of graphics in display and has played an important role in the field of mainstream computing. The original design of GPUs has long been developed into a more powerful card that even non-graphics applications can use it. The development of the graphics card continues, allowing more powerful processors to be built. The CPU communicates with the GPU through a graphics connector: PCI Express or AGP slot in the motherboard. The connector becomes responsible of transferring all commands and data from the CPU to the GPU and has evolved alongside the GPU itself for the pastyears. The original AGP slot, 32 bits wide, ran at 66 MHz, with a transfer rate of 264 MB/sec. As years go by, transfer rate doubles each available bandwidth, from 2x to 8x. In 2004, the PCI Express standard has reached a maximum bandwidth of 4GB/sec, simultaneously available to and from the GPU [5].

The current GPU can perform concurrent floating point operationsby having hundreds of processors. These processors have onchip memory of 128MB to 4GB which is accessed by the GPU with a faster rate than the CPU accessing the system memory (RAM). The GEForce 8 GPU architecture achieves up to 70GB per second memory transfers when used in some applications, relatively higher than the ordinary system memory interface which resulted to 6.4GB per second.

Disaster Relief in Laguna: A Geographical Information System Through Crowdsourcing on Facebook

Karen Anne S. Tolentino and Joseph Anthony C. Hermocilla

***Abstract*— In the Philippines, disasters happen unexpectedly. A need for better and faster reporting to the government and to the people concerned is high. Social media makes information widespread by providing tools that lets people easily share information to the public. Statistics show that Filipinos use Facebook as their main social networking site. Given the problem and the technology, an application is created to fulfill the need for a coordinated system during disasters. The application combines the power of Google Maps, Facebook, and the concept of Crowdsourcing.**

I. Introduction

A. Background of the study

The Philippines, an archipelago consisting of 7,100 islands, is considered to be a disaster-prone country because of its geographic location. Situated along the western part of the Pacific Ring of Fire, the country is surrounded by volcanoes and earthquake generators. Mount Pinatubo, a dormant volcano, shocked the country when in erupted in 1990 and caused a devastating state to the country. The Philippines also lies in the path of turbulent typhoons. Considering the archipelagic nature of the country and other problems in the country like deforestation, improper waste management, and other developmental factors, flooding is intensified and causes more trouble. Like when Ondoy struck the country last 2009, the storm dumped heavy rains which in turn caused landslide and sent floods and muds across Metro Manila and mostly parts of Luzon. Ondoy brought loss of millions of dollars in properties and affected the daily lives of Filipinos due to the unsettled floods which lasted for more than a month [**?**] [**?**] [**?**]. Aside from natural calamities, there are also human-initiated disasters like sudden fires in the city caused by careless individuals. Most of these fires happen among squatter areas where the houses are attached to each other, exaggerating the fire even more.

In the past, news about these disasters is spread out through television, newspaper, and radio. Since the evolution of internet, circulation of news is increased. Furthermore, the development of social media sites made information propagation widespread. In contrast to the regular media where we can only read a news story written on a newspaper, listen to a report aired on the radio, or watch an important event shown on television, social media allows us to do all of that with added abilities to communicate, share, connect and interact. Antony Mayfield (2008) categorized social media into seven kinds, namely, *social networks, blogs, wikis, podcasts, forums, content communities,* and *microblogging* [**?**]. A list of top social networking sites in the Philippines for the year 2010 is ranked and Facebook got the first place [**?**].

According to Gao, Barbier, and Goolsby (2011), "Social media has recently played a critical role in natural disasters as an information propagator that can be leveraged for disaster relief." One example is after the Haiti earthquake where people posted their personal experiences of the earthquake via Twitter, Flickr, Facebook, and Youtube. This benefited Red Cross which received US$8 million in donations [**?**].

B. Statement of the Problem

Although the manner of social media propagation through simple posting may bring a positive impression on the disaster, it does not provide a coordinated system for disaster relief. Its hard to visualize the whole disaster or its severity only through reading random status or post. To address these coordination issues, crisis-management applications were developed in other countries. These applications use crisis maps to better grasp the situation by visualizing the data they gathered from social media.

There is a need for that kind of system here in the Philippines to aid the country during times of disasters.

C. Significance of the Study

Basically, this study creates a Geographic Information System (GIS), which is defined by ESRI as an information system that integrates hardware, software, and data to capture, manage, display, and analyze geographic information [**?**]. By developing a GIS for the country, which makes use of the crowdsourcing power of social media and the mapping applicability of Google Maps, it would locate disasters on the map and also plot resources urgently needed. Based on the visualizations on the map, organizations and volunteers can easily determine which areas are most affected. Government and NGOs can focus on these areas while the public can avoid these areas. Facilitation and expedition of resource distribution process will also be efficient and effective as the map will show resources which are most needed and in which area.

Presented to the Faculty of the Institute of Computer Science, University of the Philippines Los Baños in partial fulfillment of the requirements for the Degree of Bachelor of Science in Computer Science

An Accounting Solution for the Open-Source Eucalyptus Cloud Computing Framework

Rodolfo N. Duldulao, Jr.
Institute of Computer Science
University of the Philippines Los Baños
rnduldulaojr@gmail.com

Chryss Ann A. Belaguin
Institute of Computer Science
University of the Philippines Los Baños
caabelaguin@gmail.com

Regelyn T. Bañacia
Institute of Computer Science
University of the Philippines Los Baños
regelyn.banacia12@gmail.com

Joseph Anthony C. Hermocilla
Institute of Computer Science
University of the Philippines Los Baños
jachermocilla@uplb.edu.ph

ABSTRACT

This research presents an accounting facility which is currently lacking from the open-source Eucalyptus cloud computing framework. The main contribution of this work is a command line tool for billing and web interface for data viewing. Utilization cost was computed based on the extracted information from virtual machine instances in the cloud. The cost rate applied was based on the pricing scheme of Amazon Elastic Compute Cloud(Amazon EC2).

Keywords

Cloud computing, Eucalyptus, Accounting, Private cloud, Amazon EC2

1. INTRODUCTION

Cloud computing is defined as "a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction". [12] It is considered an IT progression from client-server computing, wherein hardware and human resource expenses are highly needed for the deployment of an IT infrastructure.

There are three service models of cloud computing: Infrastructure-as-a-Service (IaaS), Software-as-a-Service (SaaS), and Platform-as-a-Service (PaaS). IaaS is a service where the user can transfer existing programs and data into the cloud in the form of a cloud server. In IaaS, the user has control over the cloud server's contents and some configuration. SaaS is a service wherein computer applications residing on a cloud infrastructure are accessed through a thin client interface such as a web browser. In SaaS, the user does not actually develop the applications being used. PaaS is a service where the user obtains access to a platform residing on the cloud infrastructure which can be used to develop and deliver applications. [12]

IaaS can be deployed as a private cloud. A private cloud consists of internal data centers of a business or other organization that are not made available to the public. [1] However, even if the cloud is not publicly available, it can be optionally commercialized within the organization using a utility business model.

Private clouds can be built by anyone. In building a private cloud environment, various tools are available. Some of these tools are OpenNebula, Nimbus, AbiCloud, vSphere and Eucalyptus. OpenNebula is an open-source framework designed to manage virtualization in distributed environment. Nimbus offers self-configuring virtual cluster support. AbiCloud is another open-source framework which supports virtual infrastructure management tools such as VirtualBox, Xen, KVM and VMware. vSphere is a commercial software for building a private cloud. Eucalyptus, which stands for "Elastic Utility Computing Architecture Linking Your Programs To Useful Systems", is a web services-based implementation of cloud computing which is available in open-source and enterprise versions.

Among the tools mentioned, Eucalyptus is one of the better frameworks because it is implemented using commonly-available Linux tools and basic web service technology, making it easy to install and maintain. Moreover, it provides an interface that is compatible with the Amazon Elastic Cloud Compute (Amazon EC2), a popular IaaS developed by Amazon which allows clients to use web service interfaces to launch virtual server instances in different operating systems. [3] It also has an open-source version which is free of charge and open to the community for improvement.

2. PROBLEM STATEMENT

Cloud providers usually set cost rates for their services. However, in the open-source version of Eucalyptus, the abil-

A CUDA-based Implementation of the Cellular Potts Model combined with Lattice-Gas Cellular Automata for Cancer Cell Growth Simulation

Lei Kristoffer R. Lactuan
Institute of Computer Science
University of the Philippines
Los Baños
lk.lactuan@gmail.com

Joseph Anthony C. Hermocilla
Institute of Computer Science
University of the Philippines
Los Baños
jachermocilla@uplb.edu.ph

Mehrdad Ghaemi
Department of Chemistry
Teacher Training University
Tehran, Iran
ghaemi@tmu.ac.ir

Amene Shahrokhi
District Health Center Shahre Ray
Vice-Chancellor for Health
Tehran University of Medical Science
Tehran, Iran
amene_shahrokhi@hotmail.com

ABSTRACT

The process in which cancer cells invade or grow is very complex. Thus researchers rely on simulation to study and understand this phenomenon. Cellular Pots Model (CPM) and the Lattice-Gas Cellular Automata(LGCA) is one model for cancer cell growth simulation. In this work, we present a Compute Unified Device Architecure(CUDA)-based implementation of CPM combined with LGCA and compare its performance in terms of execution time to a CPU-based implementation.

Keywords

Cellular Potts Model, Lattice-Gas Cellular Automata, cancer, cancer cells, necrotic cells, GPU, CUDA, CPU, matrix, necrosis, apoptosis, mitosis, quiescent, proliferation, model, mathematics, biology

1. INTRODUCTION

Mathematics and Biology seem to have no intersections since both disciplines are thought to be far-fetched from each other. Mathematics focuses on the numerical data while Biology focuses on complex data which is nonnumeric. However, mathematics is a powerful tool in which behavioral patterns of complex biological phenomena can be modeled using mathematical equations. This is mainly due to the expansion of both disciplines and collaborations between them, as well as because of the large amount of data accumulated from each discipline.

Cancer is a complex process since genetic changes within the infected cells occur in a sub cellular level [6]. This results into mutations within the cells that cause continuous abnormal divisions and cause destruction on neighboring cells. This manner of cell mutation and destruction is called *cancer cell growth*. Cancer has been marked as one of the diseases that is very dangerous due to its high mortality rate.

The process in which cancer cells invade or grow within a matrix of cells is complex. This makes fully understanding it, a great challenge for researchers. Since the need for fast evaluation and analysis of such complex phenomena currently arises, it attracted interest among scholars to create models for it, particularly carcer cell growth, for use in simulations.

Simulations are imitations of real objects or phenomena which does not necessarily require the actual object being simulated to be present. It is a useful tool in executing these phenomena in real time even if the phenomena being observed is out of season. Some objects that are being simulated are volcanic erruptions, tsunamis, and other disasters. Animal or insect behavior can also be simulated. Some of these animals or insects are bees, fishes, and more.

Simulations, particularly in disasters, allows one to observe what will happen during a disaster given specific parameters. Thus, we can plan ahead on how to prevent the disaster or plan on how to escape from it. This is also applicable on biological simulations wherein one can observe how diseases spread throughout the body as well as how one might know how to treat the disease.

Through the years, a lot of models concerning cancer growth have been developed, simulated, and implemented; such models show cell-to-cell interactions and competitions. This pa-

A CUDA-based Implementation of the Cellular Potts Model combined with Lattice-Gas Cellular Automata for Cancer Cell Growth Simulation

Lei Kristoffer R. Lactuan
Institute of Computer Science
University of the Philippines
Los Baños
lk.lactuan@gmail.com

Joseph Anthony C. Hermocilla
Institute of Computer Science
University of the Philippines
Los Baños
jachermocilla@uplb.edu.ph

Mehrdad Ghaemi
Department of Chemistry
Teacher Training University
Tehran, Iran
ghaemi@tmu.ac.ir

Amene Shahrokhi
District Health Center Shahre Ray
Vice-Chancellor for Health
Tehran University of Medical Science
Tehran, Iran
amene_shahrokhi@hotmail.com

ABSTRACT

The process in which cancer cells invade or grow is very complex. Thus researchers rely on simulation to study and understand this phenomenon. Cellular Pots Model (CPM) and the Lattice-Gas Cellular Automata(LGCA) is one model for cancer cell growth simulation. In this work, we present a Compute Unified Device Architecure(CUDA)-based implementation of CPM combined with LGCA and compare its performance in terms of execution time to a CPU-based implementation.

Keywords

Cellular Potts Model, Lattice-Gas Cellular Automata, cancer, cancer cells, necrotic cells, GPU, CUDA, CPU, matrix, necrosis, apoptosis, mitosis, quiescent, proliferation, model, mathematics, biology

1. INTRODUCTION

Mathematics and Biology seem to have no intersections since both disciplines are thought to be far-fetched from each other. Mathematics focuses on the numerical data while Biology focuses on complex data which is nonnumeric. However, mathematics is a powerful tool in which behavioral patterns of complex biological phenomena can be modeled using mathematical equations. This is mainly due to the expansion of both disciplines and collaborations between them, as well as because of the large amount of data accumulated from each discipline.

Cancer is a complex process since genetic changes within the infected cells occur in a sub cellular level [6]. This results into mutations within the cells that cause continuous abnormal divisions and cause destruction on neighboring cells. This manner of cell mutation and destruction is called *cancer cell growth*. Cancer has been marked as one of the diseases that is very dangerous due to its high mortality rate.

The process in which cancer cells invade or grow within a matrix of cells is complex. This makes fully understanding it, a great challenge for researchers. Since the need for fast evaluation and analysis of such complex phenomena currently arises, it attracted interest among scholars to create models for it, particularly carcer cell growth, for use in simulations.

Simulations are imitations of real objects or phenomena which does not necessarily require the actual object being simulated to be present. It is a useful tool in executing these phenomena in real time even if the phenomena being observed is out of season. Some objects that are being simulated are volcanic erruptions, tsunamis, and other disasters. Animal or insect behavior can also be simulated. Some of these animals or insects are bees, fishes, and more.

Simulations, particularly in disasters, allows one to observe what will happen during a disaster given specific parameters. Thus, we can plan ahead on how to prevent the disaster or plan on how to escape from it. This is also applicable on biological simulations wherein one can observe how diseases spread throughout the body as well as how one might know how to treat the disease.

Through the years, a lot of models concerning cancer growth have been developed, simulated, and implemented; such models show cell-to-cell interactions and competitions. This pa-

9th National Conference on I.T. Education (NCITE 2011)
Palawan State University, Puerto Princesa, Palawan
20 – 23 October 2010

Terra: A 3D Terrain Generator and Visualizer

Richard Christiensen E. Aluning
Institute of Computer Science
College of Arts and Sciences
University of the Philippines Los Baños
College 4031, Laguna, Philippines
rcaluning@yahoo.com

Joseph Anthony C. Hermocilla
Institute of Computer Science
College of Arts and Sciences
University of the Philippines Los Baños
College 4031, Laguna, Philippines
jachermocilla@uplb.edu.ph

ABSTRACT

Artificial terrains are widely used in animations, computer games, and simulations. This study investigates the characteristics of terrains generated by three algorithms namely Fault Formation, Midpoint Displacement, and Perlin Noise. Implementation of these algorithms in Terra allows users to visualize and navigate over the generated terrain in 3D.

1. INTRODUCTION

Terrain is an area of land. Examples are rocky mountains, grass plains, rolling hills, all combining to form a beautiful landscape. Terrains are used in diverse fields, not just because of their aesthetic value, but also for scientific and entertainment applications.

Modern computer games, such as flight simulators and role-playing type games, involve players navigating over mountains and islands. These types of terrains add realism to the gaming experience of players.

Scientists studying hydrological processes use terrains to test flow routing algorithms. Different types of terrain can affect the behaviour of flows. Flood simulations use terrains to determine areas where floods are most likely to occur.

Terrain data can be obtained by manual surveying of actual land formations or remote sensing. However, this approach is tedious and not cost-effective. An alternative is to use computer algorithms to generate artificial terrains for use in applications described above.

This study investigates the characteristics of terrains generated by Midpoint Displacement, Fault Formation, and Perlin Noise. These algorithms are implemented in Terra. Users can generate terrains by selecting a desired algorithm. Generated terrains can be visualized and navigated in 3D.

The succeeding sections discuss the details of the algorithms and the actual result of the implementation in Terra. Sample terrains for each algorithm were generated and evaluated.

2. THEORETICAL FRAMEWORK

2.1 Heightmaps

A heightmap is used to store elevation data. It is normally represented as a grayscale image file or a two-dimensional array of values. Algorithms for generating artificial terrains populate a heightmap with height or elevation values.

2.2 Midpoint Displacement

Midpoint Displacement[2], also known as Plasma Fractal or Diamond Square Algorithm, is a fractal technique which exhibits self-similarity and is recursive. A heightmap's corners are named points A, B, C and D. The midpoint of segments AB(1), BD(2), DC(3) and CA(4) are first calculated. The midpoints (1,3) and (2,4) are then connected to obtain the intersection point E. To calculate the height of E, the height values of A, B, C, and D are averaged and a $2^{-roughness}$ coefficient is added. Each quadrant is then processed recursively using the same principle. Figure 1 illustrates how Midpoint Displacement works.

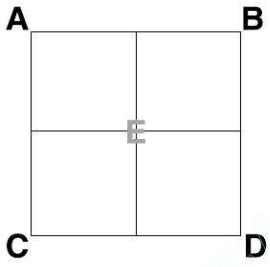

Figure 1: Initial configuration of a heightmap when using Midpoint Displacement.

2.3 Fault Formation

Fault Formation[7] generates faults in a terrain. A random line is added to an empty heightmap then a random height is assigned to a side. The process is repeated until the desired terrain is achieved. The random height is linearly decreased and must eventually reach zero. Figure 2 shows the first step of Fault Formation on an initially empty heightmap.

2.4 Perlin Noise

Perlin Noise adds up noise functions at a range of different scales. It is calculated using n dimension. A noise function

Identifying Catchment Areas near Selected Mountains in the Philippines using FlowViz

Joseph Anthony C. Hermocilla
Institute of Computer Science
College of Arts and Sciences
University of the Philippines Los Baños
College 4031, Laguna, Philippines
jachermocilla@uplb.edu.ph

Jaderick P. Pabico
Institute of Computer Science
College of Arts and Sciences
University of the Philippines Los Baños
College 4031, Laguna, Philippines
jppabico@uplb.edu.ph

ABSTRACT

This paper presents FlowViz, a software tool for modeling, simulation, and visualization of surface water flow. FlowViz was used in identifying possible catchment areas near selected mountains in the Philippines to locate places where flood might occur.

1. INTRODUCTION

Disaster management is the discipline of dealing with and avoiding risks and involves preparing for disaster before it occurs [10]. These disasters may be natural or human-made and can cause damage to property and human life. One such disaster is flood, which is an overflow or accumulation of an expanse of water that submerges land [11]. There are several factors that cause flood including excessive surface runoff due to storms and typhoons. Floods result to physical damage and casualties, contamination of water supplies, food shortage, and other devastating effects [11]. Although majority of floods are caused by natural factors and can thus be difficult to predict, planning ahead can be done, given relevant information, in order to minimize the damage they may cause.

Modeling, simulation, and visualization are essential computational tools that can be used to generate information needed for planning. These computational techniques accept data input and produce some output based on the values of the model parameters and simulation runs. These outputs can then be used for decision-making and planning. Models can be validated by comparing the results of the simulation with actual observations, if available.

This paper presents FlowViz, a software developed to model the surface water flow over a wide area based on digital elevation models. Being able to model the surface water flow will allow the identification of possible catchment areas where flood might occur in the future. The succeeding sections describe the data input and output, flow model, software architecture, experiments, and simulation results.

2193	2233	2245	2242	2228	2200
2254	2296	2316	2315	2289	2252
2312	2361	2389	2382	2343	2279
2338	2399	2445	2414	2363	2301
2336	2379	2406	2395	2351	2298
2307	2353	2364	2327	2296	2258

Figure 1: Example DEM raster (in meters).

Although there are existing applications similar to FlowViz, classified under Geographic Information System (GIS) [12] , most of them are commercial and will require extra cost to customize. In addition, due to the integrated architecture of these existing systems, they cannot handle large data sets for wide area simulations.

2. DATA INPUT AND OUTPUT

The primary input to FlowViz is the Digital Elevation Model (DEM). A DEM is a digital representation of ground surface topography or terrain [9]. DEMs are generated using remote sensing in order to cover larger geographical area, although traditional surveying can still be used. They are described using a grid or matrix with values in each cell representing elevation data. DEMs can vary in resolution. The resolution pertains to the total area covered by the elevation value. Figure 1 shows an example of a DEM in the ESRI ASCII Grid Format [6].

Another input used by FlowViz is the precipitation data. Precipitation data is described by the Probability of Precipitation (POP). It is a formal measure of the likelihood of precipitation used in weather forecasting [8]. In FlowViz, precipitation data is represented also as a grid with the amount of precipitation specified in each cell. There is a one-to-one mapping between the DEM grid and the precipitation grid, that is, each elevation cell has a corresponding precipitation cell. Depending on the given POP, a precipitation cell may

A Multiagent System Framework for Solving the Student Sectioning Problem

[Extended Abstract]

Joseph Anthony C. Hermocilla
Institute of Computer Science
University of the Philippines Los Baños
College 4031, Laguna, Philippines
jachermocilla@uplb.edu.ph

Eliezer A. Albacea
Institute of Computer Science
University of the Philippines Los Baños
College 4031, Laguna, Philippines
eaalbacea@uplb.edu.ph

ABSTRACT

Student sectioning is the assignment of students to classes in such a way that no classes assigned to a student conflict in schedule and no class exceeds a specified class size. This paper proposes a multiagent system framework for solving the Student Sectioning Problem.

Keywords

student sectioning, multiagent systems, algorithms

1. INTRODUCTION

Student sectioning is the assignment of students to classes in such a way that no classes assigned to a student conflict in schedule and no class exceeds a specified class size. It is an important problem that a university must address when automating its student registration process, especially in universities with large number of enrollees and classes. The student sectioning problem is usually treated as a subproblem of the more general timetabling problem.

We define the Student Sectioning Problem as a tuple $SSP = (A, B, C, D)$ where A is a set of students with elements a, B is a set of subjects with elements b, C is a set of classes with elements a pair $c = (b, section)$, and D is a set of $write-in$ with elements $d = (a, b)$. We specify the attribute *timeslot* to a class c. We define *slot* as a pair $t = (c, n)$ and we let E be the set of all slots. The $classize(c)$ is the number of slots with c in the elements of E. We define an *assignment* as a pair $f = (d, t)$ such that given a write-in d and the slot set E, $(b, section)$ is in C and c is in t. We also define a predicate $conflict(a, b)$ over a set of assignments Q such that given any two assignments f_1 and f_2 in Q, it returns *true* if the timeslots of the c in f_1 and c in f_2 are the same and $false$ otherwise. The predicate $full(c)$ over a set of assignments Q returns *true* if the number of assignments in Q which include c is greater than $classsize(c)$ .The solution to a SSP is a set S of assignments such that for all students a in A, the subset X_a of S containing all assignments for student a, $conflict(f_{1a}, f_{2a})$ is $false$ and for all c in S, $full(c)$ is $false$. We refer to X_a as the *schedule* of of student a. The union of all X_a for all a in A is the set S. The *classlist* for a class c is a subset of A such that there is an assignment of student a in class c in S.

The Student Sectioning Problem can be formulated as the standard Constraint Satisfaction Problem(CSP) in artificial intelligence. A CSP is a tuple $CSP = (V, U, W)$ with a set of variables V, domain set U, and a set of constraints W. A solution to a CSP is a set of assigment of values to variables with little or no violation of constraints. Thus, standard algorithms for solving CSP's, like backtracking, can be used to solve the Student Sectioning Problem.

In this paper, we present a multiagent system framework for solving the Student Sectioning problem. We model the student registration process as a multiagent system composed of autonomous agents that exhibits specific behavior to achieve their desired goals. The emergent interaction of the agents generate a solution to the Student Sectioning Problem. The main advantage of this approach is that the assignment can be done in parallel and in a distributed manner since each agent is autonomous having its own thread of execution and can be geographically dispersed.

2. METHODOLOGY

In this framework, we defined three types of agents namely *scheduler agent*, *enlister agent*, and *student agent*. These agents are representative of the actors that interact in the student registration process in a typical university. Agent communication is accomplished via *send*() and *receive*() primitives.

2.1 Scheduler Agent

The scheduler agent is the manager agent representative of the registrar. It bootstraps the enlister and student agents and responds to the queries from student agents (requesting initial schedules). It also collects the *schedule* from each student agent. Only one instance of the scheduler agent exists in the framework. The scheduler agent is responsible for collecting the final solution to the SSP.

ICS-OS: A Kernel Programming Approach to Teaching Operating System Concepts*

Joseph Anthony C. Hermocilla
Institute of Computer Science
College of Arts and Sciences
University of the Philippines Los Baños
College 4031, Laguna, Philippines
jachermocilla@uplb.edu.ph

ABSTRACT

Traditional approaches to teaching operating systems require students to develop simulations and user space applications. An alternative is to let them modify parts of an actual operating system and see their programs run at kernel space. However, this is difficult to achieve using modern real-world operating systems partly because of the complex and large source code base. This paper presents ICS-OS and the experiences and results of using it for teaching an undergraduate operating systems course. ICS-OS is based on the DEX-OS kernel which has a smaller source code base compared to mainstream operating systems, making it ideal for instruction. The students were able to demonstrate a deeper understanding of how a real operating system works by their succesful implementation of projects to enhance and extend ICS-OS.

Categories and Subject Descriptors

D.4.7 [**Operating Systems**]: Organization and Design ; K.3.2 [**Computer and Information Science Education**]: Computer Science Education

Keywords

Operating systems, computer science education, kernel

1. INTRODUCTION

Operating systems is a core knowledge area in computer science education emphasized by the ACM and IEEE review task force for the computer science curriculum. Traditional approaches to teaching operating systems to undergraduates, as in the case at the author's institution, do not involve programming the components of an actual operating system that can run on real hardware. Instead, simulations are used or user space application development are done. Students, however, are more interested in writing code that runs at the kernel space, internal to the operating system itself. They want to concretize the abstract concepts in operating systems through kernel code. To achieve this, either the students can build an operating system from scratch[5] or modify an existing one.

Building an operating system from scratch is not the best option since a course is usually offered for a semester and there is not enough time to finish. In addition, the prerequisite knowledge needed to make an operating system may have not been acquired by the students yet. To write an operating system from scratch, one has to have knowledge of the processor architecture, assembly language, data structures, algorithms, and low-level C programming.

Modifying an actual operating system that runs on real hardware is a more viable alternative. However, the choice of the operating system to use is still an issue. In the past, several instructional operating systems have been proposed and developed. The next section briefly reviews some of them to highlight their strengths and weaknesses.

Two possible criteria for choosing the operating system to use are completeness and size of the source code base. An instructional operating system that does not implement high level abstractions like process management, memory management, and filesystems will unlikely be a good choice because of the missing features. On the other hand, an operating system with several thousands of lines of code and a complicated source directory structure will confuse students and will take more time to understand. Thus there should be a right balance between completeness and code size.

Recent developments in hardware emulation and virtualization have also made it easier to work with real-world operating systems. Testing a kernel need not require a reboot of the development machine for testing. Unnecessary boostrapping is no longer needed since the test machine is a software application running on the development machine itself.

The delivery of an operating systems course is usually through a lecture and a laboratory component. A popular textbook used by instructors in the lecture is the dinosaur book by Silberschatz and Galvin [10]. Typical laboratory activities involves learning to use a Unix-based operating system, developing simulations for different process scheduling algo-

*ICS-OS is an open source project hosted at http://code.google.com/p/ics-os/

7th National Conference on I.T. Education (NCITE 2009)
Capitol University, Cagayan de Oro City, 21-23 October 2009

Hindawi Publishing Corporation
International Journal of Plant Genomics
Volume 2008, Article ID 369601, 6 pages
doi:10.1155/2008/369601

Research Article

The Generation Challenge Programme Platform: Semantic Standards and Workbench for Crop Science

Richard Bruskiewich,[1] Martin Senger,[1] Guy Davenport,[2] Manuel Ruiz,[3] Mathieu Rouard,[4] Tom Hazekamp,[4] Masaru Takeya,[5] Koji Doi,[5] Kouji Satoh,[5] Marcos Costa,[6] Reinhard Simon,[7] Jayashree Balaji,[8] Akinnola Akintunde,[9] Ramil Mauleon,[1] Samart Wanchana,[1,10] Trushar Shah,[2] Mylah Anacleto,[1] Arllet Portugal,[1] Victor Jun Ulat,[1] Supat Thongjuea,[10] Kyle Braak,[2] Sebastian Ritter,[2] Alexis Dereeper,[3] Milko Skofic,[4] Edwin Rojas,[7] Natalia Martins,[6] Georgios Pappas,[6] Ryan Alamban,[1] Roque Almodiel,[1] Lord Hendrix Barboza,[1] Jeffrey Detras,[1] Kevin Manansala,[1] Michael Jonathan Mendoza,[1] Jeffrey Morales,[1] Barry Peralta,[1] Rowena Valerio,[1] Yi Zhang,[1] Sergio Gregorio,[1,11] Joseph Hermocilla,[1,11] Michael Echavez,[1,12] Jan Michael Yap,[1,12] Andrew Farmer,[13] Gary Schiltz,[13] Jennifer Lee,[14] Terry Casstevens,[15] Pankaj Jaiswal,[15] Ayton Meintjes,[16] Mark Wilkinson,[17] Benjamin Good,[18,19] James Wagner,[18,19] Jane Morris,[16] David Marshall,[14] Anthony Collins,[7] Shoshi Kikuchi,[5] Thomas Metz,[1] Graham McLaren,[1] and Theo van Hintum[20]

[1] *Crop Research Informatics Laboratory, International Rice Research Institute (IRRI), DAPO Box 7777, Metro Manila, Philippines*
[2] *Crop Research Informatics Laboratory, International Maize and Wheat Improvement Center (CIMMYT), Apdo. Postal 6-641, 06600 Mexico, DF, Mexico*
[3] *Centre International de Recherche Agronomique pour le Développement (CIRAD), Avenue Agropolis, 34398 Montpellier, Cedex 5, France*
[4] *Bioversity International, Via dei Tre Denari 472/a, 00057 Maccarese (Fiumicino), Rome, Italy*
[5] *National Institute for Agrobiological Sciences (NIAS), Kannondai 2-1-2, Tsukuba, Ibaraki 305-8602, Japan*
[6] *Empresa Brasileira de Pesquisa Agropecuaria (EMBRAPA), Parque Estação Biologia Final W5 Norte, 70770-900 Brasilia, DF, Brazil*
[7] *Centro Internacional de la Papa (CIP), Avenida La Molina 1895, La Molina, Apartado Postal 1558, Lima 12, Peru*
[8] *International Crops Research Institute for the Semi-Arid Tropics, Patancheru, Andhra Pradesh 502324, India*
[9] *International Center for Agricultural Research in the Dry Areas, P.O. Box 5466, Aleppo, Syria*
[10] *National Center for Genetic Engineering and Biotechnology, 113 Thailand Science Park, Phahonyothin Road, Klong 1, Klong Luang, Pathumthani 12120, Thailand*
[11] *Institute of Computer Science, College of Arts and Sciences, University of the Philippines, Los Baños, Laguna 4031, Philippines*
[12] *Department of Computer Science, University of the Philippines, Room 215, Melchor Hall, Diliman, Quezon City 1101, Philippines*
[13] *National Center for Genome Resources, 2935 Rodeo Park Drive East, Santa Fe, NM 87505, USA*
[14] *Scottish Crop Research Institute, Invergowrie, Dundee DD2 5DA, Scotland, UK*
[15] *Department of Plant Breeding, Cornell University, Ithaca, NY 14853, USA*
[16] *African Centre for Gene Technologies, P.O. Box 75011, Lynnwood Ridge 0040, South Africa*
[17] *Department of Medical Genetics, Faculty of Medicine, The University of British Columbia, Vancouver, BC, Canada V6T 1Z3*
[18] *School of Computing Science, Simon Fraser Universtiy, 8888 University Drive, Burnaby, BC, Canada V5A 1S6*
[19] *Bioinformatics Graduate Program, Genome Sciences Centre, BC Cancer Agency, 100-570 West 7th Avenue, Vancouver, BC, Canada V5Z 4S6*
[20] *Centre for Genetic Resources, The Netherlands (CGN), P.O. Box 16, 6700 AA Wageningen, The Netherlands*

Correspondence should be addressed to Richard Bruskiewich, r.bruskiewich@cgiar.org

Received 22 September 2007; Accepted 14 December 2007

Recommended by Chunguang Du

The Generation Challenge programme (GCP) is a global crop research consortium directed toward crop improvement through the application of comparative biology and genetic resources characterization to plant breeding. A key consortium research activity is the development of a GCP crop bioinformatics platform to support GCP research. This platform includes the following: (i) shared, public platform-independent domain models, ontology, and data formats to enable interoperability of data and analysis flows within the platform; (ii) web service and registry technologies to identify, share, and integrate information across diverse, globally dispersed data sources, as well as to access high-performance computational (HPC) facilities for computationally intensive,

Perceived Social Loafing in Undergraduate Software Engineering Teams

Jaderick P. Pabico*, Joseph Anthony C. Hermocilla,
John Paul C. Galang and Christine C. De Sagun
Institute of Computer Science
University of the Philippines Los Baños
{jppabico,jachermocilla,jpcgalang,ccdesagun}@uplb.edu.ph

ABSTRACT

We surveyed 237 undergraduate students who are enrolled in various subjects and are members of software engineering teams. Their being a member in a team is part of the requirements of the course. We found that each of task visibility, distributive justice, and intrinsic task involvement were negatively associated with social loafing. We also found out that dominance, aggression and sucker effect each were positively correlated with social loafing. We further found out that perception of social loafing exists among members of software engineering teams.

Keywords

software engineering, teams, social loafing, free riding, task visibility, contribution, distributive justice, dominance, sucker effect.

1. INTRODUCTION

In most management of classes in the Philippines, working in teams allows students to participate in the active construction of knowledge, enhance problem solving skills, share ideas and opinions, gain valuable experience, and learn lessons regarding group communications and problem solving that can be used later in real-world work environment. These are the reasons listed by the proponents of group work as a tool for learning in most educational systems [2, 3, 10]. In most information technology and computer science courses, this instruction technique is usually employed by instructors in courses that require the students to learn to work within a team, such as those that teach the management and control of software production teams. One of the difficulties in assessing students' output in a team is that the individual contribution is sometimes indistinguishable from the team's output [30]. One of the reasons for this is that without the proper tools and technology, the individual's progress is difficult to track. When a student perceives that her[1] inputs to the team will not be given due recognition, either by her team members or by her instructor, her motivation to contribute to the team will be low [15]. Both the benefits of claiming high grades due to high levels of effort and the penalties of getting low grades due to low levels of effort in contributing to the team have become non-factor in motivating the students to perform better [15]. Because of this perception, the productivity of the team diminishes.

One well-researched reasons for productivity losses in teams is the tendency of a team member to decrease her effort when working in a team compared to when she is working alone [31]. This tendency is called *social loafing* [20] and has been shown to occur in various team tasks. Work done by various researchers suggest that the occurrence of social loafing in teams are due to perceived behavioral factors of the team members. These researchers, however, have studied social loafing exclusively in laboratory settings where the environment is controlled for easy measurement of the variables. Others have conducted their research in ongoing teams, but under real-world job settings and cultures different from that of the Philippines (see for example [6] and [9]). So far, none have studied on-going teams in the classroom settings in the Philippines. Thus, our research sought to increase understanding and awareness of social loafing as it occurs in the classroom settings and as influenced by local Filipino culture. More specifically, we wanted to determine whether the perception of social loafing and other behavioral factors exist in the minds of students who are members of undergraduate software engineering teams. We tested hypotheses concerning the effect of the perceived behaviors of team members to

*Corresponding author

[1]Please note that we used the female gender as our writing style only, and not as a means to prejudice the opposite gender.

6th National Conference on I.T. Education (NCITE 2008)
University of the Philippines, Los Baños, 23-24 October 2008