

## Research Article

### Development of LiDAR Database Management System using Open Source Software

<sup>1</sup>Khairil Izwan Ahmad Arshad, <sup>1,2</sup>Helmi Z.M. Shafri, <sup>1,2</sup>Shattri Mansor and <sup>3</sup>Raja Azlina Raja Mahmood

<sup>1</sup>Department of Civil Engineering,

<sup>2</sup>Geospatial Information Science Research Centre (GISRC), Faculty of Engineering,

<sup>3</sup>Faculty of Computer Science and Information Technology, Universiti Putra Malaysia (UPM), 43400 Serdang, Selangor, Malaysia

**Abstract:** This study is focused on the development of a LiDAR database management system for the Department of Survey and Mapping Malaysia (JUPEM) to facilitate user authentication, retrieval of LiDAR datasets, storage, scheduling of LiDAR flight sessions and generation of new data products. Additionally, the design goal of the data management system is to support managers-vendors relationship, as well as new data generation out of the results. In this study, we described the structure development of LiDAR database management system and how the system collaborated with data production and data acquisition. The architecture of such application in WebGIS, providing map interactivity in displaying a simple dataset of LiDAR data by using Open Layers integration via GeoJSON format as the spatial data were used to show the implementation of such feature. We also used the Open Source Software (OSS) in the development of the LiDAR Database Management System for JUPEM. This was because many countries, especially in the public sector, were slowly transferring from using proprietary software to OSS.

**Keywords:** CodeIgniter, GeoJSON, MVC, openlayers, open source software, web 2.0: LiDAR, webGIS

## INTRODUCTION

The Department of Survey and Mapping Malaysia (JUPEM) was established in 1885 and became one of the earliest agencies in Malaysia that performs survey and mapping activities. Later in 1965, the government approved the setting up of "Direktorat Pemetaan Negara Malaysia" (Directorate of National Mapping, Malaysia) that was responsible in surveying, mapping, topographic and geodetic activities. Since then, JUPEM has commenced into various modernizations and restructuring programs in order to improve its products and services. This effort continues on in the era of satellite technology. Like any modern land management entity, JUPEM uses LiDAR (Light Detection and Ranging) as a technology in producing high-resolution maps with the applications in geomatics and contour mapping.

Malaysia's economy has grown by a minimum of 6% per annum with the private sector being the main contributor to economic growth. Private investment grew by 7.3% per annum (Economic Planning Unit, 2014). The booming in development has demanded a big survey data, which is why the Malaysian Government has imposed a 1/2007 Circular of

Geospatial Safety Instructions on The Classified Document to control the rapid and large captured data. The Malaysia Survey and Mapping Department (JUPEM) as a stake holder of data mapping, annually receives vast amount of LiDAR data due to the requirement that all mapping data collected by any bodies or companies must deliver a copy to JUPEM as the data library (section 2 number 14 (V)).

LiDAR has been widely applied in various applications (Cao *et al.*, 2015). LiDAR is a rapidly developing measurement technology that has been extensively applied in feature detection (Cheng *et al.*, 2013), terrain monitoring (Zhou and Vosselman, 2012), 3D model reconstruction (Cheng *et al.*, 2013) and ground object extraction (Kumar *et al.*, 2013). The Aerial Laser Scanning (ALS) has been used to collect topographic data points for a large area, which triggers million points of cloud data to be acquired (Cao *et al.*, 2015). These LiDAR point cloud generally contain millions of points that support for the good management of these data that are still in their infancy (Mosa *et al.*, 2012).

The large size of LiDAR topography data causes many organizations to struggle in managing the relatively large size of datasets (Nandigam *et al.*, 2010).

**Corresponding Author:** Helmi Z.M. Shafri, Department of Civil Engineering, Faculty of Engineering, Universiti Putra Malaysia (UPM), 43400 Serdang, Selangor, Malaysia, Tel.: +60-3-89466459

This work is licensed under a Creative Commons Attribution 4.0 International License (URL: <http://creativecommons.org/licenses/by/4.0/>).

No existing integrated framework solution can exploit on the accessing and storage of these vast data sets such as the one like LiDAR (Lewis *et al.*, 2010). User only wants a simple and easy method to store them, a quick and simple solution to simply save the data and is understandable (Bjørset, 2013).

However, there might be benefits from having the LiDAR data stored in a more organized manner. Point clouds are versatile and can be re-used many times for different purposes. This becomes more interesting when the acquisition cost is taken into account. Usually the raw LiDAR data are collected by a variety of vendors and the datasets may also be available in different formats, with different schemes, different attributes and different compression schemes.

There are cases where the data were recorded in different geographic projections, even within the same dataset (Nandigam *et al.*, 2010). The processed point clouds are then written into output files of different formats, with the preferred format being the open LAS file format (ASPRS, 2008). LAS format is currently a common way of storing and managing LiDAR point cloud (ASPRS, 2010).

There are various ASCII formats that exist, usually as XYZ file, XYZI or extended version with intensity. There are also ASCII columns and positioned in any possible order. The most usual features are the pulse number, GPS acquisition time, or classification field (David *et al.*, 2008). Additionally, a file can represent different things, a tile of LiDAR data or a strip (raw acquisition geometry). There are some formats that propose spatial indexing (McGaughey, 2009). The modern LiDAR library should not only be able to manage these standards but also to provide functionalities that extended ASCII for export and import in ensuring the compatibility with the current software packages and toolkits (David *et al.*, 2008).

Currently, LiDAR data are stored either in an American Society for Photogrammetry and Remote Sensing (ASPRS) LAS format or an ASCII format (often without a standard definition of the format) which has been the standard for storing discrete return LiDAR data as a binary file (ASPRS, 2010). Many GIS and statistical analysis programs face difficulties in working with the LAS file. These programs often do not have the capabilities to write and read LAS file format (Ehinger, 2010).

Generally, after they have served their purposes, it can be time-consuming and resource demanding to store these point cloud that contain millions or even billions of points in a constructive way. Usually, all these data sets end up in file, external hard drives or other storage solutions (Bjørset, 2013).

A file can represent different things; a strip (raw acquisition geometry) or a tile of LiDAR data. Some formats also propose spatial indexing (McGaughey, 2009). It becomes clear that a modern LiDAR library

should be able to manage these two standards but also to supply the extended ASCII import and export functionalities for ensuring compatibility with toolkits and current software packages (David *et al.*, 2008).

Even though the intention to organize LiDAR data is there, the direction in selecting a system that can manage the data is problematic. Software solutions for storing and working with LiDAR data are still in their preliminary stage. As a result, many current spatial databases and Geographical Information Systems have poor support for LiDAR data and these data management solutions cannot be used straight away (Bjørset, 2013).

The current obstacle of JUPEM's organization and management of LiDAR data would be on these three grounds:

**No centralized data archival and retrieval:** Since the data received from some of the Vendors are in CD, archiving data would require a lot of manual work to copy and organize the data in a hierarchical or historical order. The challenges soon follows with the retrieval of the data since there is no centralized system that prevents data from overlapping or have the ability to compare out-dated data vs. updated data.

This would move unnecessary human resources if the retrieval of the data is crucial e.g. on rescue mission, national security, etc.

**No automatic data acquisition from vendor:** Since JUPEM deals with a lot of vendors to retrieve raw data from the ground, automation would be a better option to consider in managing large datasets as well as managing many vendors.

Thus, automation is needed to ensure data retrieval can be monitored via a central dashboard as well as keeping track on users' request from either the vendor (in supplying data) or from the public (in demanding data).

**Interactive map:** Most of the map representations are still in physical format and its digital representations are currently managed on data-points only. With the help of Open Layers and centralized data points in one database, we can optimize the usage of both to provide the user with a better graphical interface as well as interactivity with the data stored.

Due to the global economic downturn, many organizations that have increasingly adopted the Open Source Software (OSS) are more practical in their perceptions and opinions regarding of the value of (OSS) (Fauscette, 2009). The increased cost of operations and decreasing cost in revenue in the current economic climate organizations caused many researcher to look for OSS as a way to reduce IT expenditure (Shaikh and Cornford, 2012; Ward and Tao, 2009).

Table 1: Benefits of open source software

Benefits	Sub-benefits
Political	Reduce software piracy vendor independence security customization
Economic	Cost saving develop local industry promote competition
Social	Develop local skills create employment digital divide
Technical	Open standards interoperability

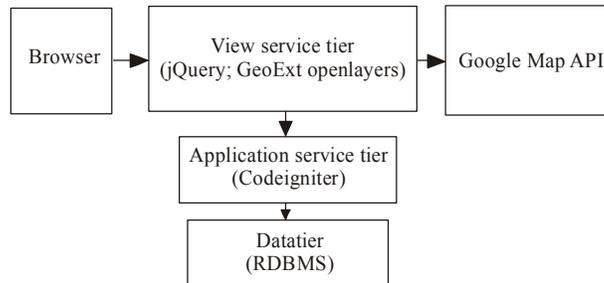


Fig. 1: Web GIS system architecture

Many governmental initiatives, research institutions, educational as well software vendors use open-source tools in the development of multiple applications. The independencies from relying on a single software entity and wider peer review of OSS (person, group or firm) are often considered secure and sustainable approaches for software development (Achuthan *et al.*, 2014).

Open Source Software (OSS) has been increasingly considered by most public sector organizations as a viable alternative to proprietary software (Holck *et al.*, 2004; McNaughton, 2010). Governments in countries such as Brazil, Russia, India, China and South Africa had widely adopted OSS (Kemp, 2009; Henley and Kemp, 2008). There is an open standard policy for citizens to access government applications that do not focus on specific vendor software in the European Union (Ghosh and Schmidt, 2006; Haider and Koronios, 2009; Hwang, 2005).

This study was conducted based on Malaysia economy situation in a year 2016, indicating the need of cost effective database management system. Therefore, this study aims to provide alternative for LIDAR database management system, particularly for government institution through the usage of OSS.

Table 1 shows the benefits of OSS from the perspective of the public sector by data group from articles published between 2003 to 2012 (Jokonya, 2015).

## MATERIALS AND METHODS

**System architecture:** The architecture is divided into three-tiers with each tier representing the essential part of data handling before the viewer. The tigers are as follows in Fig. 1:

**View service tier:** View Service Tier is responsible to organize data to be represented to the viewer at the

browser. This tier displays feature points, basic operations of a map (navigation, zoom, etc) and basic management of information.

**Application service tier:** Application Service Tier is responsible for getting the data from the Data tier to be re-organized as GeoJSON format and feed it to the View Service Tier. It also handles the user request e.g., different features points based on a state, country, vendor datasets, etc.

This tier manages the interchange-ability of the data as well as doing basic data management e.g. similar datasets are stored in a cache so that it will not burden the Data Tier with many requests (which slows down response time).

**Data tier:** Data tier is the repository tier for all the stored data of any given datasets samples. It resides in the relational database structure and is tasked to withhold the data integrity.

### Tiers system structure:

**System software resources:** This study used XAMPP v1.8.305 for setting up the local prototype server. XAMPP comes with the web server Apache 2.4.10, PHP 5.5.15 for server-side programming language and MySQL 5.6.20 for database language.

XAMPP also comes with phpMyAdmin 4.7.2.1 for a simple database web system that helps to manage the database setup.

For the User Interface (UI), this study used Bootstrap v3.3.0 (<http://getbootstrap.com>) which is robust enough to cater clean and responsive UI. Bootstrap v3.3.0 comes with jQuery for client-side needs.

For Map Interactivity, this study used OpenLayers v3.0.0 (<http://openlayers.org>) that cater to most of the requirements to fetch and display LiDAR data points and provides the basic map interactivity. It works well

with jQuery on Bootstrap and thus requires minimal JavaScript library to work.

**Application structure-MVC (Codeigniter) and WebGIS (Openlayers):** MVC is a Pattern-Oriented Software Architecture that was initially developed for Graphical User Interface (GUI) purposes by TrygveReenskaug during his visit to Xerox Parc to introduce MVC in Smalltalk-76 in 1970. It later became a common way to design systems by segregating between business rules (data, logic and rules of the application) and output to the users.

MVC concept consists of three distinct components that govern significant responsibilities. The components are:

**Model:** Which is the core of the components, governs the behaviour of the system? In Model, 'business rules' take shapes and from these rules, it will define how data can be governed. By business rules, it meant the logic of what data can be manipulated and how it should be manipulated. By manipulation, it meant how data (or links between data) can be created, updated, removed or viewed by the users.

**View:** Governs the output of a data determined by the model. View can be passive where it only shows the output from the Model or it can be active where it manipulates the output given by the Model.

**Controllers:** On the other hand are where the data is given behaviour. By behaviour, it meant how data is given by the Model that is represented to the View.

MVC is separated from each other and run independently, so in this way it made systems development work relatively independently, greatly improving the efficiency of the development of the system and strengthen the maintainability of the program (Hao *et al.*, 2015).

**Codeigniter:** An open-source development framework that uses MVC to enable user to deploy websites in a short amount of time. However, it does not strictly uses MVC since it allows the users to do much of the procedural programming in the Controller part.

It was developed initially by EllisLab as part of the Expression Engine core in 2006 and was recently transferred to the British Columbia Institute of Technology for further development on 2014.

It was developed on PHP and has a quite lean learning curve, thanks to the good documentation as well as supportive community and architecture. The latest version as of date is version 3.0-dev but the researchers used the version 2.2 since it has an established track record and have strong community support amongst the web development framework.

WebGIS is an application that caters interoperability, heavily focused on the UI/UX and

interactivity between systemic biases. The interactivity is made possible with the emergence of client-side/web browser technologies such as Asynchronous JavaScript and XML (AJAX), Ajax frameworks e.g., YUI, jQuery, ExtJS, etc which are the cornerstone of Web 2.0 achievement in providing seamless interaction of the user with the server-side processes without having to reload the page after each call to action.

AJAX is able to use a special format of XML-like structure in order to communicate a large dataset. This special format is called JavaScript Object Notation (JSON). Since both formats (XML and JSON) are comprehensible in JavaScript, a developer can easily use them to transmit structured data in the web application without much page request (which are sometimes too time consuming). Ajax makes web pages and data exchange between servers what makes web pages become more efficient and stable (Hao *et al.*, 2015).

In JSON, an object is an unordered set of name/value pairs. The object is encapsulated with both left and right braces e.g., { }. Names inside the object is followed by a colon and the name/value pairs are separated by comma and usually written with both name and value padded by a double-quote, e.g., { "type": "Feature", "geometry": "Multipoint" }.

An array is an ordered collection of values. An array is encapsulated with left and right brackets, [ ] and its values are separated by comma, e.g. [101.9, 2.9]

The LiDAR data were set as the JSON object that follows certain structure, usually known as GeoJSON. GeoJSON uses the JSON structure as the basic architecture that is lightweight and easy for humans to read and write as well as easy for machines to parse and generate. The object within the architecture follows a geographic format to ease the interchange-ability of providing and receiving the LiDAR data.

GeoJSON is built on two structures:

- The first is a collection of name/value pairs. In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or an associative array.
- The second is an ordered list of values. In most languages, this is realized as an array, vector, list, or sequence.

This data structure is constructed to generate feature list, create objects and string recursively in the list. This allows the developer to create geometry, properties object and coordinates array.

OpenLayers is a JavaScript library package designed for Web GIS client development and it is used to implement map data access and is published by the standard format. This javascript library has been developed to further the usage of geographic information of any kinds. OpenLayers is completely free, released under a BSD-style License and currently

has the support for OGC WMS layers, navigation, icons, markers and layer selection which also supports XML, GML, GeoJSON, JSON, etc interchange format in order to boost interchange-ability for any web platform. This also means it can bind any other mapping services data sources provided by other company e.g. Google Maps or Microsoft, as long as the format is supported.

It uses the Ajax technology to browse map page without refreshing the browser. In addition to providing the general map browsing functions, OpenLayers supports WFS (Web Feature Service), WMS (Web Mapping Service) and other network services specification, which are developed by the Open GIS association. The map data to the OpenLayers client follows the OGC (Open GIS Consortium) standard which renders the display function based on the capability of the browser.

Additionally, OpenLayers not only supports zoom, pan and other common operations, but also supports in selecting a face, selecting a different operating line, feature selection, layer stack and other operations. In the field of interactive intelligent power information, it is important to first achieve the load map information in order to be able to show the power of information.

**Database organization:** MySQL database tables contain the information necessary to authenticate users, record file ownership, construct search functions for processing LiDAR datasets, schedule flight dates, determine file sharing permissions for teams of users and locate datasets.

The data are separated into 2 dataset groups. The groups are Users and Areas. User's dataset group comprises of all user-related tables that governs its authentication, identity and roles relationship. The Areas dataset group comprises of all LiDAR data tables that govern the files' ownership, relationship with area data and flight details.

**Table's description:** For Users dataset:

- Users: All information of the individual users
- Groups: List of roles of a user
- Users\_groups: relationships between users and groups (roles)
- Login\_attempts: attempts of login by a user.
- Vendor: Company related information that is associated with a particular user.

For Areas dataset:

- Flight\_list: all information of flight data and tracks upload/process date.
- Classification: give human readable classification for a classification ID.
- Area: all information given in .las files that's being processed

**Archival ability:** Archiving is done in two stages. The first stage would be archiving all the files when the users/managers upload their LiDAR files. The second stage would be processing the data in the database where it provides search ability within the system.

**Retrieval ability:** The database is designed so that during a user's web session, session id and username are remembered. Any admin group member can archive datasets that can achieve users. The interfaces of web page for different roles will display only the data products that were created by the user. The user has the option to discard the results and redo the analysis with different inputs or to store generated product in the database for future use.

The supplementary information stored within foreign keys relating derived products to process inputs and tables can be used to achieve the specific data, in addition to retrieving the data products by permissions or username. There are functions which have the foreign key relating Flight list and areas that have an overlap to the Users' datasets. Users can also track all datasets that used the overlap function if they wish or retrieve datasets based on Flight name, user/ company or by location name.

**Data products:**

**Flight data:** The File\_list table contains all relevant information of the flight session. This includes:

**Flight date and location:** Where locations are specified by State in Malaysia and the districts of that particular state.

**File data:** File\_name, file\_size, uploaded by whom and date on when the file is uploaded and being processed.

**LiDAR data:** The Areas dataset contains LiDAR data that is separated as follows:

- X: the x-axis of a LiDAR point
- Y: the y-axis of a LiDAR point
- Z: the z-axis of a LiDAR point
- Intensity: the intensity of data for a point in x, y, z axis
- Return\_num:
- Num\_of\_returns:
- Scan\_dir\_flag:
- Edge\_flight\_line: flights' boundary that has been covered in a given file
- Scan\_angle\_bank: the angle of x, y, z axis
- Classification: the types of structure that exist in the x, y and z axis are associated with the classification table.
- User\_data:
- Point\_source\_id:
- Gps\_time: time of when this point is taken in GPS format.
- Status: the status of this particular point.

With these fields, the managers are able to specify what kind of data is needed when they search for a particular set of data combination. For example, the managers are able to specify whether the data they need for retrieval should include Intensity or GPS time of any given flight session that is located in Maran, a district of Pahang state in Malaysia.

**Implementation:** The system used [maps.opengeo.org](http://maps.opengeo.org) WMS web map as the map provider and uses relational database to supply the coordinate data to the viewer via the GeoJSON format.

In this study, in order to simplify the implementation, pre-set coordinates were used to show how the user will be presented with points for a Malaysian state. The datasets are populated using an estimation of various points that exist in the Data Tier.

#### **Generating GeoJSON:**

- From the Data Tier, a set of coordinates are generated in x, y points e.g., [101.2, 3.8] where the first data (left) is the x value of the coordinate and the second data (right) is the y value of the coordinate. It is encapsulated within a bracket.
- Coordinate data is then parsed to a series of bracket divided by comma e.g., [101.2, 3.8], [101.2, 3.5], [101.5, 3.2].

These Coordinate data is then packed into a GeoJSON template suitable to be a feed to the OpenLayers standard.

**Controlling the map:** Used the zoom in/out button situated at the top-left corner of the map. It is denoted with +/-button. The user is also able to browse the map by dragging the map across.

**View the datasets in the Map:** Change the `_smap_selangor.js` in the `ol.source`. GeoJSON to `_smap_sarawak.js`

**Data conversion:** The software chosen is the BCAL Software as the LiDAR data conversion from the las file to ASCII file that is the BCAL Software. The demand for free and open source LiDAR processing software is increasing as more LiDAR datasets are available and the price of new data acquisitions decreases (Ehinger, 2010). BCAL LiDAR is a processing tool that has been developed at the Boise Center Aerospace Labrotary (BCAL) of Idaho State University and it helps in filling this need. The software, written in the Interactive Data Language (IDL), has point data analysis capabilities including subsetting, filtering, classification and rasterization.

The tools, referred to as BCAL LiDAR, automate common processing tasks, such as file format, height filtering, conversion, rasterization and subsetting. BCAL LiDAR software was used to convert LiDAR in the LAS file to an ASCII file due to the fact that it has a fullwave format conversion and has been used widely by researchers.

## **RESULTS**

Figure 2 and 3 shows the result of file data conversion. The size after conversion for both low and high accuracy area revealed an unexpected result. Both have shown an almost exponential conversion size but with a bit of anomaly for the low density data sample. A comparison between both data showed that for the low accuracy data, the different size of the LAS file and ASCII was about 2.86 times bigger but for the high accuracy file, it was only 2.57 times bigger. We could say that performance-wise, it satisfied our expectation by showing that both have shown size expansions after being converted to ASCII from LAS.

The reason for the almost exponential size expansion was because LAS file is a compacted zipped file that has header files on top of the data points. Converting them to ASCII unzips them and the data-points are arranged in a series of rows and column. The rows and column are necessary for the data-points to be read via Comma Separated Value (CSV). CSV is a popular format that can easily transfer the data-points to the MySQL database and this serves the purpose of why it's being converted from LAS to ASCII and later to CSV.

For a visual guide, Fig. 4 shows the data uploads time for various set ups. Each legend indicates the processor-bandwidth data uploaded against time. Performance-wise, all processor-bandwidth suggested that the larger the file, the longer it is needed to be uploaded to the server. However, Core 2 Duo-10Mbps performed better in comparison to Pentium 4 and i7 processors. i7-5Mps would be the most consistent of all and it indicated the performance benchmark we wanted.

However, other processor-bandwidth combination suggested that at some point, there was an anomaly in uploading the data. The most unexpected result was the i7-10Mbps combo that performed badly compared to Core 2 Duo (for both bandwidths) and was at par with the Pentium 4's uploading time. We can only speculate that during the time of upload, there were other process (es) being run (either locally or via internet) at the same time, thus disrupting the uploading process.

We could safely conclude that the i7 processor (which represented the vendor machine) had a reasonable performance range since vendors sometimes uploads data from various locations. As for the Core 2

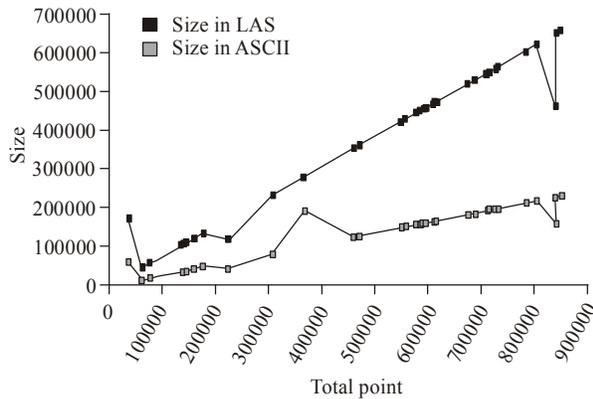


Fig. 2: Implementation result-LAS to ASCII conversion, size vs. total points for low accuracy data taken in Pekan, Pahang

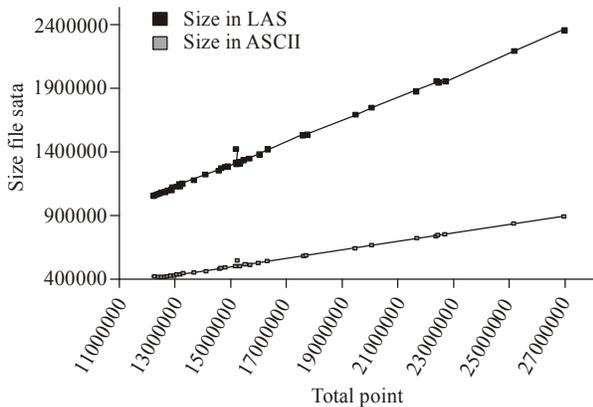


Fig. 3: Implementation result-LAS to ASCII conversion, size vs. total points for high accuracy data taken in Kuala Lumpur

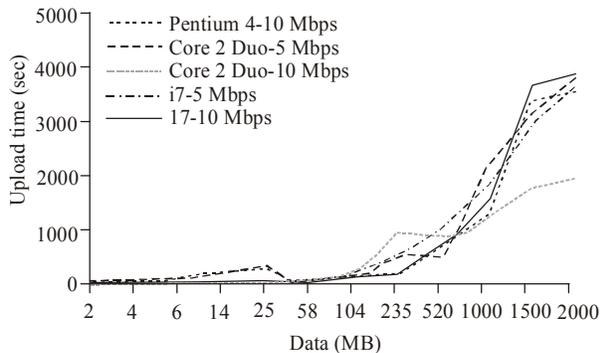


Fig. 4: LiDMS deployment performance result

Duo's and Pentium 4's (representing manager and admin respectively) performances, both have shown us that various performances did exist, despite being desktops (as opposed to being mobile) and this can be further corrected by dedicating both machine to file conversions and upload user files instead of running it for other processor-intensive task(s). However, as the Core 2 Duo machines have shown us, it performed way better than any other processor-bandwidth combination,

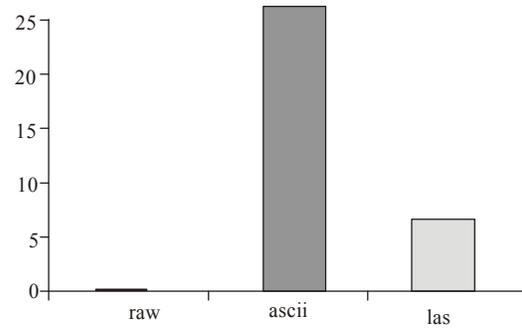


Fig. 5: Percentage of LiDAR Data file type requested

thus making it an ideal benchmark against the future manager's machine.

In Fig. 5 show that In Malaysia, more people tend to have data in the XYZ format which is an ASCII file rather than LAS file due to their needs and usages, especially in the government and public sectors. There are only 20% of the combination ratio of academicians and the public sector that needed data in LAS file. The usage of ASCII type data needed more comparison and it was easy to use for most of their current available software.

This is because we can see that public users are only interested in data reflecting search in the map. That could be the reason for the ASCII type file's popularity amongst the others, based on the questionnaire for the user sample. The ASCII type file is popular since most of the users only need simple data e.g., xyz points. Hence, it is easier for them to get a bulk load of data and make their conclusion from it.

As for the LAS type file, it is widely known that vendor and students favour it due to the LAS type file that has the heading that holds intrinsic data e.g., flight date, date, summary, etc as well as the much smaller file size. Even though some students may only need similar data that are available in ASCII, students might prefer LAS since they are more familiar with the software that handles LAS file better. It would be more convenient for them to have LAS file since they might get better results with the software they are using, e.g., the software are able to translate points to 3D view from LAS file, able to get niche data, etc.

We can easily conclude that most of the users prefer to have 'xyz' data only, rather than all (or xyz+ intensity). This shows that the basic expectation of the users to this web application has been fulfilled since we captured almost all of the LiDAR data in our database. This has also been reflected via our search and export capability. We can say that the upgrade needed in the future for the search and export capability is to be able to handle xyz data more thoroughly (e.g., by having a different table just for x, y and z values) and our system need to improve in archiving LiDAR data in different variants (i.e., xyz, xyz+I or other combinations).

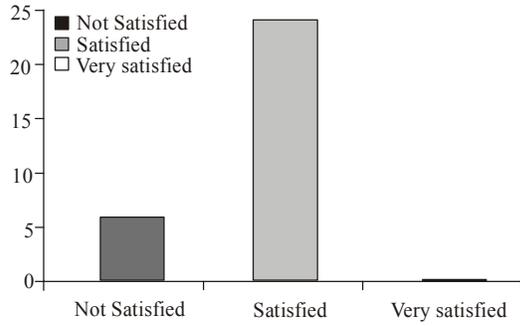


Fig. 6: Performance percentages of work efficiency

With the usage of ENVI BCAL, we were able to convert raw data from the LAS format to ASCII. From the LiDAR ASCII format, we were able to port them to Comma Separated Value (CSV) that was readable and storable in the MySQL database. Conversion was also done and although this made the converted file larger, it served the purpose in storing them in a more manageable manner in the Web system.

While the conversion portion was being done, the Web system can be designed and implemented to manage the uploads of the files based on a common and popular CSV format. The designs incorporated better manageability for the user of the system and the data in the system.

From there, when the data was finally uploaded and stored, we used OpenLayers as the interactivity provider. WebGIS was able to be implemented as a Web 2.0 application. The data-interchange format, GeoJSON facilitated the data transfers between Data Tier to View Service Tier via Application Service Tier.

The system can also connect to other geographic services (e.g., arcGIS, geoserver), as long as the data generated was able to be parsed in the correct GeoJSON format.

## DISCUSSION

The results from the tests and questionnaires showed that the system performed at a normal level of usage in this version. It would satisfy the entry level needed for the managers, vendors and public users. This study also showed that regardless of the machine tested, the web-based management system was able to handle the request steadily but was unable to perform with many concurrent requests. This was understandable since we were working on a local-based and limited capability machine. This can be improved if we were able to scale it further with a better machine that had the proper specifications.

Figure 6 show the percentages of the overall respondents about the effectiveness of Portal LiDMS V 1.0, where 83% of the respondents said that they were satisfied using the Portal and 17% of them said that

they were very satisfied. This study showed that none of them said they were not satisfied at all. Most of the answers were how their work became easier and faster in terms of understanding about the LiDAR data that JUPEM have.

Most of the satisfied users suggested that the portal could be more feasible if it can provide interactivity at some places. For example, the export function could have a feature where it can tell the expected downloading period. Another suggested that the generated map could have a popup that tells the user on the details of building type, name, vendor name, date of flight, etc.

LiDMS had proven to be a useful and necessary tool for the managers and vendors at JUPEM, providing them with a better management tool to compile data and manage inputs from various sources.

Relying on open-source software extended the capability of easy deployment where the LiDMS setup took less than an hour or so regardless of the operating system chosen. But it bore significance in setting up the LiDMS in a Linux environment which proved to be the better choice since it was not only fast but had the potential to be scalable in the online version. This was because most of the web server hosted for online services was Linux-based (apart from its popularity and massive community support). However, the robustness of the system being able to set up in different operating environment proved that the system is the way forward for the managers and vendors in Malaysia. Apart from managers and vendors at JUPEM, using web-based open-source solutions would also benefit the end users as well, since web-technologies are expanding beyond the traditional desktop-based applications.

Although there are many improvements that can be made in terms of delivery speed and performance as well as search capability, the system served its initial purpose which is to provide management tools for the managers and vendors in an integrated framework as well as the search and export function for the general public users. The feedback from the satisfied users signalled that the objective of the system had been met and there are more room for improvements. This is essential for the LiDAR manageability prospect in the future, especially within the governmental institutions. However, it should not be limited to LiDAR manageability, but should also be expanded so that all interconnected institutions within or essential to JUPEM can be managed via the emerging modern web-based technologies which could reduce backlogs and improve interactivity between managers, vendors and clients to LiDAR data.

Table 2 shows previous studies done on LiDAR data management. Based on the issues and results from previous findings, it is clear that choosing the best LiDAR end file format is still a problem, hence correlate to the need from Malaysian user.

Table 2: Related studies

Sources	Contribution	Issue
A LiDAR Collaboratory Data Management System (Sharma and Parikh, 2006)	MySQL as database and MATLAB as file system application	Data format in MPL-NASA binary produced by software supplied by the LiDAR vendor in a binary NetCDF format.
Library Concept And Design For LiDAR Data Processing (David <i>et al.</i> , 2008)	Modern LiDAR library should be able to manage and supply extended ASCII format	No practical or sample database created. Just as a concept.
Mobile Mapping system LiDAR data framework (Lewis <i>et al.</i> , 2010)	PostgreSQL database solution with PostGIS spatial extensions	Does not comprehensively define a complete DBMS LiDAR data handling solution
Database Design for High-Resolution LIDAR Topography Data (Nandigam <i>et al.</i> , 2010)	Datasets using IBM's DB2 database.	Hosted by San Diego Supercomputer Centre (SDSC) and sponsored by the National Science Foundation (NSF) with a big budget. Database size 8.5 times bigger than LAS file.
Sorted Pulse Data (SPD) Format: A new file structure for storing and processing LiDAR data (Bunting <i>et al.</i> , 2011)	Develop a new LiDAR format that focuses on efficient storage called Hierarchical Data Format version 5.0 (HDF5)	New format not recognized by major player.
LIDAR Datavarehus (Bjørset, 2013)	PostgreSQL database solution with PostGIS point cloud table database.	PostGIS database for point cloud has poor storage efficiency. Database size 4.5 bigger than LAS file.

Having MVC as the web-framework, we can expect that the management system be improved robustly since each branch (model, view and controller) can be improved at their respective branch in their own pace. For example, if the View of the management system needs improvement (say, to allow better user experience by having more simplified user flow or mobile version where it can be browsed from smart phones or tablets), then the View can be outsourced to a more capable expert in improving the User experience (UX). If the Model needs to be improved on handling multiple databases or improved table structure, it can be developed in the Model section alone without (or with minor update) the Controllers or the View section of the MVC.

Therefore, it is a recommended approach in developing this system since it offers scalability and improvements between different branches at their own urgency and at their own types of improvement.

### CONCLUSION

Related to the effort in providing a sustainable management system to JUPEM officers, it could be perceived that adopting web-based implementation to archive data enables it to be deployed in both standalone machines as well as online. Not limited to the deployment capability, open source software helps to ease raw data conversion to be database-ready. Web-based open-source platform would be the platform of the future for the management to cater ever-expanding geo-spatial data due to good community support. The platforms already include fundamental base that would enable the future of data management in the Big Data era. Benefits that JUPEM is able to reap with this study are:

The emergence of rapid web-deployment frameworks such as MVC helps manage the system's deployment. MVC helps in providing a sustainable development of the system by having different branches for the 'business rules/data logic', behaviour and output.

This study uses MVC to provide a prototype that would serve as a base for JUPEM to be able to move forward in improving the development of each different branch at their own pace.

This study also shows that JUPEM respondents prefer ASCII files and open source software such as BCAL ENVI and XAMPP as they could muster a variety of tasks including conversion from LAS to ASCII (regardless of data accuracy, in relatively linear time), data archiving, management tools for managers and simple yet informational interface for researchers to obtain data.

It also contributes towards data model design and promotes web-based management system. This in turn promotes additional features of a Web-GIS system, particularly in the Malaysian scene using geospatial data available in Malaysia. There is a huge area of improvement which involves the system's scalability and niche improvisations. Having open source as the base development helps due to the support availability, up to date communities and frequent feedbacks and lower set up costs.

Having most of the data aspect organized online would be cost effective since managers and vendors can work together at their own pace within a specific timeline and those who have used old medium.

Managers are able to manage vendors and data files from each data session, giving them the advantage of planning better data retrieval by having the ability to understand vendor's behavior, data quality and other measures. All of the information can be analyzed and tracked. For example, vendors can submit their data online and managers can keep track of each vendor's LiDAR maps, thus enabling them to direct the vendor's next task/assignment. Managers and higher authorized personnel have the ability to get updates on the latest map based on the information available from the system.

With the improvement of the processing capability, it is hoped that huge file problems will be a thing of the past. Thanks to the large community of open-source solutions, the management can adopt improvisations that are available online at their discretion.

## ACKNOWLEDGMENT

The authors would like to extend their heartfelt gratitude to the LiDAR unit, JUPEM Kuala Lumpur for providing the LiDAR data and Company AkarniagaSdnBhd, a company based in Cyberjaya for lending their server for the authors to do the test upload and download data online.

## REFERENCES

- Achuthan, K., S. SudhaRavi, R. Kumar and R. Raman, 2014. Security vulnerabilities in open source projects: An India perspective. Proceeding of the 2nd International Conference on Information and Communication Technology (ICoICT, 2014), Bandung, Indonesia, pp: 18-23.
- ASPRS (American Society for Photogrammetry and Remote Sensing), 2008. American society for photogrammetry and remote sensing annual conference 2008: Bridging the horizons: New frontiers in geospatial collaboration. Proceeding of a Meeting. Portland, Oregon, Held at 28 April-2 May 2008, 2: 836.
- ASPRS (American Society for Photogrammetry and Remote Sensing), 2010. LAS Specification Version 1.3–R11. Retrieved from: [http://www.asprs.org/wp-content/uploads/2010/12/LAS\\_1\\_3\\_r11.pdf](http://www.asprs.org/wp-content/uploads/2010/12/LAS_1_3_r11.pdf). (Accessed on: October 24, 2015)
- Bjørset, K., 2013. LiDAR data warehouse. M.A. Thesis, Norwegian University of Science and Technology, Trondheim, Norway.
- Bunting, P.J., J. Armston, R. Lucas and D. Clewley, 2011. Sorted Pulse Data (SPD) Format: A new file structure for storing and processing LiDAR data. SilviLaser 2011, Hobart, Australia, Oct. 17-19.
- Cao, V.H., K.X. Chu, N.A. Le-Khac, M.T. Kechadi, D. Laefer and L. Truong-Hong, 2015. Toward a new approach for massive LiDAR data processing. Proceeding of the 2nd IEEE International Conference on Spatial Data Mining and Geographical Knowledge Services (ICSDM, 2015). Fuzhou, pp: 135-140.
- Cheng, L., L. Tong, Y. Chen, W. Zhang, J. Shan, Y. Liu and M. Li, 2013. Integration of LiDAR data and optical multi-view images for 3D reconstruction of building roofs. *Opt. Laser. Eng.*, 51(4): 493-502.
- David, N., C. Mallet and F. Bretar, 2008. Library concept and design for lidar data processing. Proceeding of the GEOgraphic Object Based Image Analysis (GEOBIA) Conference, Calgary, Canada.
- Economic Planning Unit, 2014. Retrieved from: <http://www.epu.gov.my/en/video-tahun-2014>.
- Ehinger, S.E., 2010. Design, development and application of lidar data processing tools. Ph.D. Thesis, Idaho State University.
- Fauscette, M., 2009. The Value of Open Source. IDC Analyst Connection. Retrieved from: [https://www.redhat.com/f/pdf/IDC\\_749\\_CarveOut\\_Costs.pdf](https://www.redhat.com/f/pdf/IDC_749_CarveOut_Costs.pdf).
- Ghosh, R.A. and J.P. Schmidt, 2006. Open source and open standards: A new frontier for economic development? UNU-MERIT Policy Brief No. 1, Maastricht, The Netherlands, pp: 1-7.
- Haider, A. and A. Koronios, 2009. Promises of open source software for Australian government agencies-an exploratory study. Proceeding of the Pacific Asia Conference on Information Systems (PACIS). Taipei, Taiwan.
- Hao, W., C. Heping and Q. Tian, 2015. Research on manufacturing SME information management systems based on Ajax and MVC. Proceeding of the 7th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA, 2015). Nanchang, pp: 884-887.
- Henley, M. and R. Kemp, 2008. Open source software: An introduction. *Comput. Law Secur. Rev.*, 24(1): 77-85.
- Holck, J., M.H. Larsen and M.K. Pedersen, 2004. Identifying business barriers and enablers for the adoption of open source software. Proceeding of the 13th International Conference on Information Systems Development. Vilnius, Lithuania.
- Hwang, S.Y., 2005. Adopting open source and open standards in the public sector: Five deciding factors behind the movement. *Mich. J. Pub. Aff.*, 2: 1-19.
- Jokonya, O., 2015. Investigating open source software benefits in public sector. Proceeding of the 48th Hawaii International Conference on System Sciences (HICSS, 2015). Kauai, HI, pp: 2242-2251.
- Kemp, R., 2009. Current developments in open source software. *Comput. Law Secur. Rev.*, 25(6): 569-582.
- Kumar, P., C.P. McElhinney, P. Lewis and T. McCarthy, 2013. An automated algorithm for extracting road edges from terrestrial mobile LiDAR data. *ISPRS J. Photogramm.*, 85: 44-55.
- Lewis, P., C. McElhinney, B. Schön and T. McCarthy, 2010. Mobile mapping system LiDAR data framework. Proceeding of the 5th International 3D GeoInfo Conference. Berlin, Germany, November 3-4, 38: 135-138.
- McGaughey, R.J., 2009. FUSION/LDV: Software for LIDAR data analysis and visualization. US Department of Agriculture, Forest Service, Pacific Northwest Research Station, University of Washington, Seattle, WA, USA, 123(2).
- McNaughton, M., 2010. 16P. Broadening the revolution: An assessment of open source initiatives in the Caribbean and Latin America. Proceeding of the International Conference on Information Resources Management (CONF-IRM). Montego Bay, Jamaica.
- Mosa, A.S.M., B. Schoen-Phelan, M. Bertolotto and D. Laefer, 2012. Evaluating the benefits of octree-based indexing for LiDAR data. *Photogramm. Eng. Rem. S.*, 78(9): 927-934.

- Nandigam, V., C. Baru and C. Crosby, 2010. Database design for high-resolution LIDAR topography data. In: Gertz, M. and B. Ludäscher (Eds.), Scientific and Statistical Database Management. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 6187: 151-159.
- Shaikh, M. and T. Cornford, 2012. Strategic drivers of open source software adoption in the public sector: Challenges and opportunities. Proceeding of the 20th European Conference on Information Systems (ECIS). Barcelona, Spain.
- Sharma, N.C.P. and J.A. Parikh, 2006. A LIDAR collaborative data management system. Proceeding of the IEEE International Conference on Geoscience and Remote Sensing Symposium. Denver, CO, pp: 817-820.
- Ward, D.J. and E.Y. Tao, 2009. Open source software use in municipal government: Is full immersion possible? Proceeding of the World Congress on Engineering and Computer Science. San Francisco, USA, Vol. 2.
- Zhou, L. and G. Vosselman, 2012. Mapping curbstones in airborne and mobile laser scanning data. *Int. J. Appl. Earth Obs.*, 18: 293-304.