Data warehouse and decision support on integrated crop big data

Vuong M. Ngo*

Ho Chi Minh City Open University,
Ho Hao Hon 35, District 1, HCMC, Vietnam
Email: vuong.nm@ou.edu.vn
Email: vuong.cs@gmail.com
*Corresponding author

Nhien-An Le-Khac and M-Tahar Kechadi

School of Computer Science,
University College Dublin,
Belfield, Dublin 4, Ireland
Email: an.lekhac@ucd.ie
Email: tahar.kechadi@ucd.ie

Abstract: The introduction of modern information technologies for collecting and processing agricultural data revolutionise the agriculture practices. The agricultural data mining today is considered a Big Data application in terms of volume, variety, velocity and veracity. Hence, it is a challenge and a key foundation to establishing a crop intelligence platform. The platform, which processes vast amounts of complex and diverse information, will enable efficient resource management and high quality agronomy decision making. In this paper, we designed and implemented a continental level agricultural data warehouse (ADW). ADW is characterised by its (1) flexible schema; (2) data integration from real agricultural multi datasets; (3) data science and business intelligent support; (4) high performance; (5) high storage; (6) security; (7) governance and monitoring; (8) consistency, availability and partition tolerant; (9) cloud compatibility. We also evaluate the performance of ADW and present some complex queries to extract and return necessary knowledge about crop management.

Keywords: data warehouse architecture; constellation schema; Hive; MongoDB; Cassandra; smart agriculture; agricultural data challenges.

Reference to this paper should be made as follows: Ngo, V.M., Le-Khac, N-A. and Kechadi, M-T. (2020) 'Data warehouse and decision support on integrated crop big data', *Int. J. Business Process Integration and Management*, Vol. 10, No. 1, pp.17–28.

Biographical notes: Vuong M. Ngo received his BE, ME and PhD in Computer Science at HCMC University of Technology in 2004, 2007 and 2013, respectively. He is currently a Computer Scientist at Universities of Vietnam and Ireland. Previously, he held positions as a CIO, Vice-Dean and Head of Department about Information Technology in Vietnam Universities. His research interests include information retrieval, sentiment analysis, data mining, graph matching and data warehouse.

Nhien-An Le-Khac is currently a Lecturer at the School of Computer Science, UCD and a Programme Director of MSc Programme in Forensic Computing and cybercrime investigation. He obtained his PhD in Computer Science in 2006 at the Institut National Polytechnique Grenoble, France. His research interest spans the area of cybersecurity and digital forensics, data mining/distributed data mining for security, grid and high performance computing.

M-Tahar Kechadi was awarded PhD and Master degrees in Computer Science from University of Lille 1, France. He joined the UCD School of Computer Science in 1999. He is currently Professor of Computer Science at UCD. His research interests span the areas of data mining, data analytics, distributed data mining, heterogeneous distributed systems, grid and cloud Computing, cybersecurity, and digital forensics. He is a Principal Investigator at Insight Centre for Data Analytics and CONSUS project. He is a member of IEEE and ACM.

This paper is a revised and expanded version of a paper entitled 'Designing and implementing data warehouse for agricultural big data' presented at the *8th International Congress on Big Data*, San Diego, CA, USA, 25–30 June, 2019.

1 Introduction

Annual world cereal productions were 2608 million tons and 2,595 million tons in 2017 and 2018, respectively (USDA report, 2018; FAO-CSDB report, 2018). However, there were also around 124 million people in 51 countries who faced food crisis and food insecurity (FAO-FSIN report, 2018). According to United Nations (UN document, 2017), we need an increase 60% of cereal production to meet 9.8 billion people's needs by 2050. To satisfy the huge increase in demand for food, crop yields must be significantly increased using modern farming approaches, such as smart farming also called precision agriculture. As highlighted in the European Commission report (EC report, 2016), precision agriculture is vitally important for the future and can make a significant contribution to food security and safety.

The precision agriculture's current mission is to use the decision-support system (DSS) based on Big Data approaches to provide precise information for more control of waste and farming efficiency, such as soil nutrients (Rogovska et al., 2019), early warning (Rembold et al., 2019), forecasting (Bendre et al., 2015), irrigation systems (Huang et al., 2013), evapotranspiration prediction (Paredes et al., 2014), soil and herbicide and insecticide optimisation (Ngo and Kechadi, 2020), awareness (Lokers et al., 2016), supply chain (Protopop and Shanoyan, 2016) and financial services (Ruan et al., 2019). Normally, the DSSs implement a knowledge discovery process also called data mining process, which consists of data collection and data modelling, data warehousing, data analysis (using machine learning or statistical techniques), and knowledge deployment (Dicks et al., 2014). Hence, designing and implementing an efficient agricultural data warehouse (ADW) is one of the key steps of this process, as it defines a uniform data representation through its schema model and stores the derived datasets so that they can be analysed to extract useful knowledge. However, currently, this step was not given much attention. Therefore, there are very few reports in the literature that focus on the design of efficient ADWs with the view to enable agricultural Big Data analytics and mining. The design of large scale ADWs is very challenging, because the agricultural data is spatial, temporal, complex, heterogeneous, non-standardised, high dimensional, collected from multi-sources, and very large. In particular, it has all the features of Big Data; volume, variety, velocity and veracity. Moreover, the precision agriculture system can be used by different kinds of users at the same time, for instance by farmers, policymakers, agronomists, and so on. Every type of user needs to analyse different information, thus requiring specific analytics.

Unlike in any other domains, such as health-care, financial data, etc, the data and its warehousing in precision agriculture are unique. This is because there are very complex relationships between the agricultural data dimensions. The data sources are very diversified and varying levels of quality. Precision agriculture (PA) warehousing has many decision-making processes and each needs different levels of data access and different needs of analysis. Finally, there are many stakeholders involved in the data ownership and exploitation. So, the data has significant number of uncertainties. For examples, the quality of data collected by farmers depends directly on their knowledge, routines and frequency of information recording, and support tools, etc. All these issues make the PA data unique when it becomes to its storage, access, and analysis. These issues may exist in other domains, but not at the same scale and as in agriculture practices.

In this research, we firstly analyse real-world agricultural Big Data to build the effective constellation schema. From this schema, some simple questions can be easily answered directly from the modelled data. These questions include:

- For a given field, what kind of crops are suitable to grow?
- Which companies can purchase a specific crop with the highest price in the past season?
- List the history of soil texture and applied fertilisers for a given field.
- List costs of production for wheat and barley in the last five years, and so on.

Secondly, the proposed ADW has enough main features and characteristics of big data warehouse (BDW). These are

- high storage capacity, high performance and cloud computing compatibility
- flexible schema and integrated storage structure
- data ingestion, monitoring, and security to deal with the data veracity. Besides, an experimental evaluation is conducted to study the performance of ADW storage.

The rest of this paper is organised as follows: in the next section, we review the related work about decision support systems and data warehouses in agriculture. In Sections 3–5, we presented big data aspects of PA, our ADW architecture and its modules. In Sections 6–9, the quality criteria, implementation, performance analysis and decision-making applications of the proposed ADW are presented respectively. Section 10 gives some concluding remarks and future research directions. Finally, a concrete example about the ADW and its operational average run-times are shown in the appendix.

2 Related work

In precision agriculture, DSSs are designed to support different stakeholders such as farmers, advisers and policymakers to optimise resources, support farms' management and improve business practices (Gutierreza et al., 2019). For instance, DSSs were built to

- manage microbial pollution risks in dairy farming (Oliver et al., 2017)
- analyse nitrogen fertilisation from satellite images (Lundstrom and Lindblom, 2018)
- control pest and disease under uncertainty in climate conditions (Devitt et al., 2017)

- manage drip irrigation and its schedule (Friedman et al., 2016)
- predict and adopt climate risks (Han et al., 2017).

However, the datasets that were used in the mentioned studies are small. Besides, they focused on using visualisation techniques to assist end-users understand and interpret their data.

Recently, many papers have been published on how to exploit intelligent algorithms on sensor data to improve agricultural economics (Pantazi, 2016; Park et al., 2016; Hafezalkotob et al., 2018; Udiasa et al., 2018) and Rupnik et al. (2019). In Pantazi (2016), the authors predicted crop yield by using self-organising-maps; namely supervised Kohonen networks, counter-propagation artificial networks and XY-fusion. In Park et al. (2016), one predicted drought conditions by using three rule-based machine learning; namely random forest, boosted regression trees, and Cubist. To select the best olive harvesting machine, the authors in Hafezalkotob et al. (2018) applied the target-based techniques on the main criteria, which are cost, vibration, efficiency, suitability, damage, automation, work capacity, ergonomics, and safety. To provide optimal management of nutrients and water, the paper Udiasa et al. (2018) exploited the multi-objective genetic algorithm to implement an E-Water system. This system enhanced food crop production at river basin level. Finally, in Rupnik et al. (2019) the authors predicted pest population dynamics by using time series clustering and structural change detection which detected groups of different pest species. However, the proposed solutions are not scalable enough to handle agricultural Big Data; they present weaknesses in one of the following aspects: data integration, data schema, storage capacity, security and performance.

From a Big Data point of view, the papers Kamilaris et al. (2018) and Schnase et al. (2017) have proposed "smart agricultural frameworks". In Kamilaris et al. (2018), the authors used Hive to store and analyse sensor data about land, water and biodiversity which can help increase food production with less environmental impact. In Schnase et al. (2017), the authors moved toward a notion of climate analytics-as-a-service, by building a high-performance analytics and scalable data management platform, which is based on modern cloud infrastructures, such as Amazon web services, Hadoop, and Cloudera. However, the two papers did not discuss how to build and implement a DW for a precision agriculture.

The proposed approach, inspired from Schulze et al. (2007), Schuetz et al. (2018), Nilakanta et al. (2008) and Ngo et al. (2018), introduces ways of building ADW. In Schulze et al. (2007), the authors extended entity-relationship concept to model operational and analytical data; called multi-dimensional entity-relationship model. They also introduced new representation elements and showed how can be extended to an analytical schema. In Schuetz et al. (2018), a relational database and an RDF triple store were proposed to model the overall datasets. The data is loaded into the DW in RDF format, and cached in the RDF triple store before being transformed into relational format. The actual data used for analysis was contained in the relational database. However, as the schemas used in Schulze et al. (2007) and Schuetz et al. (2018) were based on entity-relationship models, they cannot deal with high-performance, which is the key feature of a data warehouse.

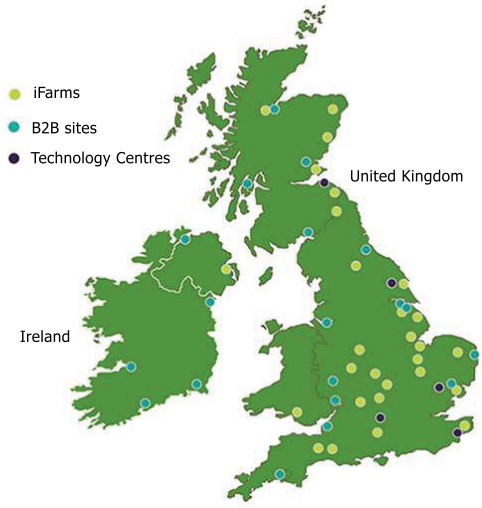
In Nilakanta et al. (2008), a star schema model was used. All data marts created by the star schemas are connected via some common dimension tables. However, a star schema is not enough to present complex agricultural information and it is difficult to create new data marts for data analytics. The number of dimensions of the DW proposed in Nilakanta et al. (2008) is very small; only three dimensions – Species, Location, and Time. Moreover, the DW concerns livestock farming. Overcoming disadvantages of the star schema, the authors of Ngo et al. (2018) and Ngo and Kechadi (2020) proposed a constellation schema for an agricultural DW architecture in order to satisfy the quality criteria. However, they did not describe how to design and implement their DW.

3 Crop big data

3.1 Crop datasets

The datasets were primarily obtained from an agronomy company, which extracted it from their operational data storage systems, research results, and field trials. Especially, we were given real-world agricultural datasets on iFarms, Business-to-Business (B2B) sites, technology centres and demonstration farms. Theses datasets were collected from several European countries and they are presented in Figures 1 and 2 (Origin report, 2018). These datasets describe more than 112 distribution points, 73 demonstration farms, 32 formulation and processing facilities, 12.7 million hectares of direct farm customer footprint and 60, 000 trial units.

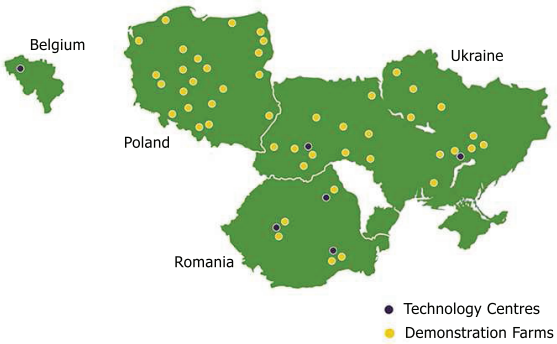
Figure 1 Data from UK and Ireland (see online version for colours)



There are a total of 29 datasets. On average, each dataset contains 18 tables and is about 1.4 GB in size. Each dataset focuses on a few details that impact the crop. For instance, the weather dataset includes information on location

of weather stations, temperature, rainfall and wind speed over time. Meanwhile, soil component information in farm sites, such as mineral, organic matter, air, water and micro-organisms, were stored in the soil dataset. The fertiliser dataset contains information about field area and geographic position, crop name, crop yield, season, fertiliser name and quantity.

Figure 2 Data in continental Europe (see online version for colours)



3.2 Big Data challenges

Raw and semi-processed agricultural datasets are usually collected through various sources: internet of thing (IoT) devices, sensors, satellites, weather stations, robots, farm equipment, farmers and agronomists, etc. Besides, agricultural datasets are very large, complex, unstructured, heterogeneous, non-standardised, and inconsistent. Hence, it has all the features of Big Data.

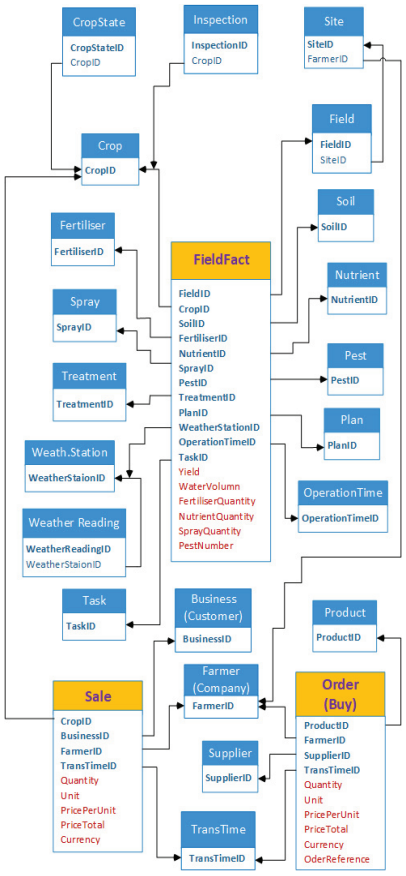
- *Volume*: The amount of agricultural data is increasing rapidly and is intensively produced by endogenous and exogenous sources. The endogenous data is collected from operational systems, experimental results, sensors, weather stations, satellites, and farming equipment. The systems and devices in the agricultural ecosystem can be connected through IoT. The exogenous data concerns the external sources, such as government agencies, retail agronomists, and seed companies. They can help with information about local pest and disease outbreak tracking, crop monitoring, food security, products, prices, and knowledge.
- *Variety*: Agricultural data has many different forms and formats, structured and unstructured data, video, imagery, chart, metrics, geo-spatial, multi-media, model, equation, text, etc.
- *Velocity*: The collected data increases at very high rate, as sensing and mobile devices are becoming more efficient and cheaper. The datasets must be cleaned, aggregated and harmonised in real-time.
- *Veracity*: The tendency of agronomic data is uncertain, inconsistent, ambiguous and error prone because the data is gathered from heterogeneous sources, sensors and manual processes.

3.3 ADW schema

The DW uses schema to logically describe the entire datasets. A schema is a collection of objects, including tables, views, indexes, and synonyms which consist of some fact and dimension tables (Oracle document, 2017). The DW schema can be designed based on the model of source data and the user requirements. There are three kind of models, namely star, snowflake and fact constellation. With the its various uses, the ADW schema needs to have more than one fact table and should be flexible. So, the constellation schema, also known galaxy schema, should be used to design the ADW schema.

We developed a constellation schema for ADW and it is partially described in Figure 3. It includes few fact tables and many dimension tables. FieldFact fact table contains data about agricultural operations on fields. Order and Sale fact tables contain data about farmers' trading operations. The key dimension tables are connected to their fact table. There are some dimension tables connected to more than one fact table, such as *Crop* and *Farmer*. Besides, *CropState, Inspection, Site*, and *Weather Reading* dimension tables are not connected to any fact table. *CropState* and *Inspection* tables are used to support *Crop* table. While, *Site* and *Weather Reading* tables support *Field* and *WeatherStation* tables. FieldFact fact table saves the most important facts about the field; yield, water volume, fertiliser quantity, nutrient quantity, spray quantity and pest number. While, in Order and Sale tables, the important facts needed by farm management are quantity and price.

Figure 3 A part of ADW schema for precision agriculture (see online version for colours)



The dimension tables contain details on each instance of an object involved in a crop yield or farm management. Figure 4 describes attributes of *Field* and *Crop* dimension tables. *Field* table contains information about name, area, coordinates (being longitude and latitude of the centre point of the field), geometric (being a collection of points to show the shape of the field) and site identify the site that the field it belongs to. *Crop* table contains information about name, estimated yield of the crop (estYield), BBCH Growth Stage Index (BbchScale), harvest equipment and its weight. These provide useful information for crop harvesting.

Figure 4 Field and Crop dimension tables (see online version for colours)

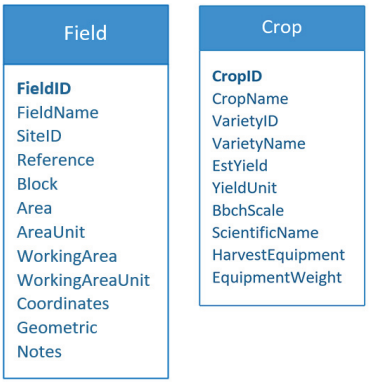
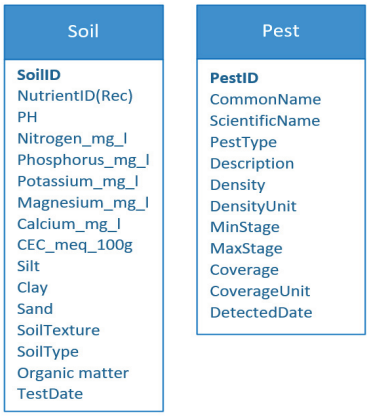


Figure 5 describes attributes of *Soil* and *Pest* dimension tables. *Soil* table contains information about pH value (a measure of the acidity and alkalinity), minerals (nitrogen, phosphorus, potassium, magnesium and calcium), its texture (texture label and percentage of Silt, Clay and Sand), cation exchange capacity (CEC) and organic matter. Besides, information about recommended nutrient and testing dates ware also included in this table. In *Pest* table contains name, type, density, coverage and detected dates of pests. For the remaining dimension tables, their main attributes are described in Table 1.

Figure 5 Soil and Pest dimension tables (see online version for colours)



4 ADW architecture

A DW is a federated repository for all the data that an enterprise can collect through multiple heterogeneous data sources; internal or external. The authors in Golfarelli and Rizzi (2009) and Inmon (2005) defined DW as a collection of methods, techniques, and tools used to conduct data analyses, make decisions and improve information resources. DW is defined around key subjects and involves data cleaning, data integration and data consolidations. Besides, it must show its evolution over time and is not volatile.

The general architecture of a typical DW system includes four separate and distinct modules; raw data, extraction transformation loading (ETL), Integrated Information and Data Mining (Kimball and Ross, 2013), which is illustrated in Figure 6. In that, raw data (source data) module is originally stored in various storage systems (e.g., SQL, sheets, flat files, ...). The raw data often requires cleansing, correcting noise and outliers, dealing with missing values. Then it needs to be integrated and consolidated before loading it into a DW storage through ETL module.

The integrated information module is a logically centralised repository, which includes the DW storage, data marts, data cubes and OLAP engine. The DW storage is organised, stored and accessed using a suitable schema defined by the metadata. It can be either directly accessed or used to create data marts, which is usually oriented to a particular business function or an enterprise department. A data mart partially replicates DW storage's contents and is a subset of DW storage. Besides, the data is extracted in a form of data cube before it is analysed in the data mining module. A data cube is a data structure that allows advanced analysis of data according to multiple dimensions that define a given problem. The data cubes are manipulated by the OLAP engine. The DW storage, data mart and data cube are considered as metadata, which can be applied to the data used to define other data. Finally, Data Mining module contains a set of techniques, such as machine learning, heuristic, and statistical methods for data analysis and knowledge extraction at multiple level of abstraction.

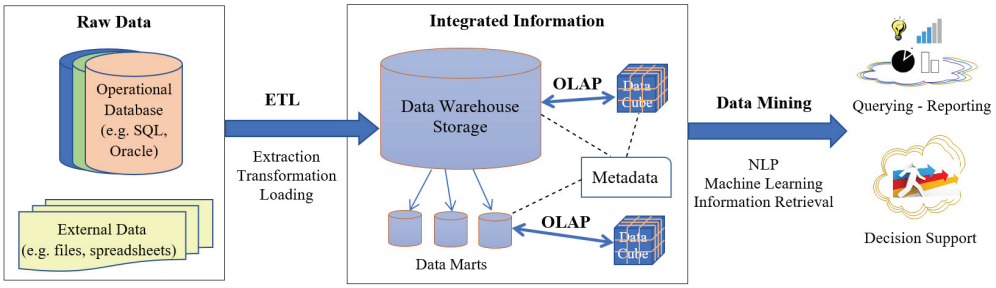
5 ETL and OLAP

The ETL module contains extraction, transformation, and loading tools that can merge heterogeneous schemata, extract, cleanse, validate, filter, transform and prepare the data to be loaded into a DW. The extraction operation allows to read, retrieve raw data from multiple and different types of data sources systems and store it in a temporary staging. During this operation, the data goes through multiple checks – detect and correct corrupted and/or inaccurate records, such as duplicate data, missing data, inconsistent values and wrong values. The transformation operation structures, converts or enriches the

Table 1 Descriptions of other dimension tables

No.	*Dim. tables*	*Particular attributes*
1	Business	BusinessID, Name, Address, Phone, Mobile, Email
2	CropState	CropStateID, CropID, StageScale, Height, MajorStage, MinStage, MaxStage, Diameter, MinHeight, MaxHeight, CropCoveragePercent
3	Farmer	FarmerID, Name, Address, Phone, Mobile, Email
4	Fertiliser	FertiliserID, Name, Unit, Status, Description, GroupName
5	Inspection	InspectionID, CropID, Description, ProblemType, Severity, ProblemNotes, AreaValue, AreaUnit, Order, Date, Notes, GrowthStage
6	Nutrient	NutrientID, NutrientName, Date, Quantity
7	Operation time	OperationTimeID, StartDate, EndDate, Season
8	Plan	PlanID, PName, RegisNo, ProductName, ProductRate, Date, WaterVolume
9	Product	ProductID, ProductName, GroupName
10	Site	SiteID, FarmerID, SiteName, Reference, Country, Address, GPS, CreatedBy
11	Spray	SprayID, SprayProductName, ProductRate, Area,Date, WaterVol, ConfDuration, ConfWindSPeed, ConfDirection, ConfHumidity, ConfTemp, ActivityType
12	Supplier	SupplierID, Name, ContactName, Address, Phone, Mobile, Email
13	Task	TaskID, Desc, Status, TaskDate, TaskInterval, CompDate, AppCode
14	Trans time	TransTimeID, OrderDate, DeliverDate, ReceivedDate, Season
15	Treatment	TreatmentID, TreatmentName, FormType, LotCode, Rate, ApplCode, LevlNo, Type, Description, ApplDesc, TreatmentComment
16	Weather reading	WeatherReadingID, WeatherStationID, ReadingDate, ReadingTime, AirTemperature, Rainfall, SPLite, RelativeHumidity, WindSpeed, WindDirection, SoilTemperature, LeafWetness
17	Weather station	WeatherStationID, StationName, Latitude, Longitude, Region

Figure 6 Agricultural data warehouse architecture (see online version for colours)



extracted data and presents it in a specific DW format. The loading operation writes the transformed data into the DW storage. The ETL implementation is complex, and consuming significant amount of time and resources. Most DW projects usually use existing ETL tools, which are classified into two groups. The first is a commercial and well-known group and includes tools such as Oracle Data Integrator, SAP Data Integrator and IBM InfoSphere DataStage. The second group is famous for it open source tools, such as Talend, Pentaho and Apatar.

OLAP is a category of software technology that provides the insight and understanding of data in multiple dimensions through fast, consistent, interactive access, management and analysis of the data. By using roll-up (consolidation), drill-down, slice-dice and pivot (rotation) operations, OLAP performs multidimensional analysis in a wide variety of possible views of information that provides complex calculations, trend analysis and sophisticated data modelling quickly. The OLAP systems are divided into three categories:

- Relational OLAP (ROLAP), which uses relational or extended-relational database management system to store and manage the data warehouse;
- Multidimensional OLAP (MOLAP), which uses array-based multidimensional storage engines for multidimensional views of data, rather than in a relational database. It often requires pre-processing to create data cubes.
- Hybrid OLAP (HOLAP), which is a combination of both ROLAP and MOLAP. It uses both relational and multidimensional techniques to inherit the higher scalability of ROLAP and the faster computation of MOLAP.

In the context of agricultural Big Data, HOLAP is more suitable than both ROLAP and MOLAP because:

- ROLAP has quite slow performance and does not meet all the users' needs, especially when performing complex calculations;
- MOLAP is not capable of handling detailed data and requires all calculations to be performed during the data cube construction;
- HOLAP inherits advantages of both ROLAP and MOLAP, which allow the user to store large data

volumes of detailed information and perform complex calculations within reasonable response time.

6 Quality criteria

The accuracy of data mining and analysis techniques depends on the quality of the DW. As mentioned in Adelman and Moss (2000) and Kimball and Ross (2013), to build an efficient ADW, the quality of the DW should meet the following important criteria:

- Making information easily accessible.
- Presenting consistent information.
- Integrating data correctly and completely.
- Adapting to change.
- Presenting and providing right information at the right time.
- Being a secure bastion that protects the information assets.
- Serving as the authoritative and trustworthy foundation for improved decision making. The analytics tools need to provide right information at the right time.
- Achieving benefits, both tangible and intangible.
- Being accepted by DW users.

The above criteria must be formulated in a form of measurements. For example, with the 8th criterion, it needs to determine quality indicators about benefits, such as improved fertiliser management, cost containment, risk reduction, better or faster decision, and efficient information transaction. In the last criterion, a user satisfaction survey should be used to find out how a given DW satisfies its user's expectations.

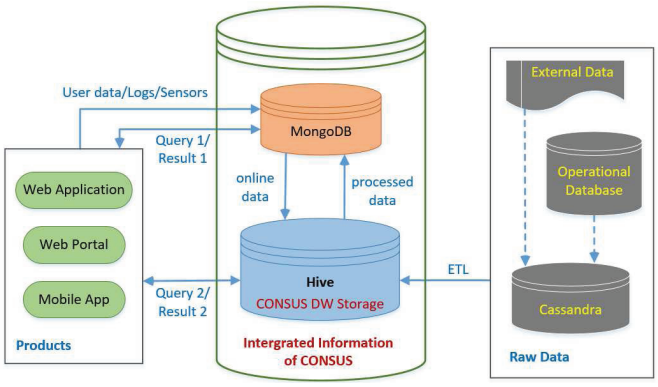
7 ADW implementation

Currently, there are many popular large-scale database types that can implement DWs. Redshift (Amazon document, 2018), Mesa (Gupta et al., 2016), Cassandra (Hewitt and Carpenter, 2016; Neeraj, 2015), MongoDB (Chodorow, 2013; Hows et al., 2015) and Hive (Du, 2018; Lam et al., 2016). In Ngo et al. (2019), the authors analysed the most popular no-sql databases, which fulfil most of the aforementioned criteria. The advantages, disadvantages, as well as similarities and differences between Cassandra, MongoDB and Hive were investigated carefully in the context of ADW. It was reported that Hive is a better choice as it can be paired with MongoDB to implement the proposed ADW for the following reasons:

- Hive is based on Hadoop which is the most powerful cloud computing platform for Big Data. Besides, HQL is similar to SQL which is popular for the majority of users. Hive supports well high storage capacity, business intelligent and data science more than MongoDB or Cassandra. These Hive features are useful to implement ADW.
- Hive does not have real-time performance so it needs to be combined with MongoDB or Cassandra to improve its performance.
- MongoDB is more suitable than Cassandra to complement Hive because:
  - MongoDB supports joint operation, full text search, ad-hoc query and second index which are helpful to interact with the users. Cassandra does not support these features.
  - MongoDB has the same master – slave structure with Hive that is easy to combine. While the structure of Cassandra is peer - to - peer.
  - Hive and MongoDB are more reliable and consistent. So the combination of both Hive and MongoDB adheres to the CAP theorem.

The ADW implementation is illustrated in Figure 7 which contains three modules, namely Integrated Information, Products and Raw Data. The Integrated Information module includes two components; MongoDB and Hive. MongoDB receives real-time data; as user data, logs, sensor data or queries from Products module, such as web application, web portal or mobile app. Besides, some results which need to be obtained in real-time will be transferred from the MongoDB to Products. Hive stores the online data and sends the processed data to MongoDB. Some kinds of queries having complex calculations will be sent directly to Hive.

Figure 7 Agricultural data warehouse implementation (see online version for colours)



In the Raw Data module, almost data in Operational Databases or External Data components, is loaded into Cassandra. It means that we use Cassandra to represent raw data storage. Hence, with the diverse formats of raw data; image, video, natural language and sql data, Cassandra is better to store them than SQL databases. In the idle times of the system, the updated raw data in Cassandra will be imported into Hive through the ELT tool. This improves the performance of ETL and helps us deploy ADW on cloud or distributed systems.

8 Performance analysis

The performance analysis was conducted using MySQL 5.7.22, JDK 1.8.0_171, Hadoop 2.6.5 and Hive 2.3.3 which run on Bash, on Ubuntu 16.04.2, and on Windows 10. All experiments were run on a desktop with an Intel Core i7 CPU (2.40 GHz) and 16 GB memory. We only evaluate the performance of reading operation as ADW is used for reporting and data analysis. The database of ADW is duplicated into MySQL to compare performance. By combining popular HQL/SQL commands, namely Where, Group by, Having, Left (right) Join, Union and Order by, we created 10 groups for testing. Every group has five queries and uses one, two or more commands (see Table 2). Moreover, every query uses operators; And, Or, $\geq$, Like, Max, Sum and Count, to express complex queries.

Table 2 Command combinations of queries

Group	*Commands*
G_1	Where
G_2	Where, Group by
G_3	Where, Left (right) Join
G_4	Where, Union
G_5	Where, Order by
G_6	Where, Left (right) Join, Order by
G_7	Where, Group by, Having
G_8	Where, Group by, Having, Order by
G_9	Where, Group by, Having, Left (right) Join, Order by
G_{10}	Where, Group by, Having, Union, Order by

All queries were executed three times and we took the average value of the their execution times. The difference in runtime between MySQL and ADW for a query q_i is calculated as $Times_{q_i} = RT_{q_i}^{mysql}/RT_{q_i}^{ADW}$. Where, $RT_{q_i}^{mysql}$ and $RT_{q_i}^{ADW}$ are average runtimes of query q_i on MySQL and ADW, respectively. Moreover, with each group G_i, the difference in runtime between MySQL and ADW is $Times_{G_i} = RT_{G_i}^{mysql}/RT_{G_i}^{ADW}$. Where, $RT_{G_i} = Average(RT_{q_i})$ is average runtime of group G_i on MySQL or ADW.

Figure 8 describes the time difference between MySQL and ADW for every query. Although running on one computer, but with large data volume, ADW is faster than MySQL on 46 out of 50 queries. MySQL is faster for three queries 12th, 13th and 18th belonging to groups 3^{rd} and 4th. The two systems returned the same time for query 24th from group 5th. Within each query group, for fair performance comparison, the queries combine randomly fact tables and dimensional tables. This makes complex queries taking more time and the time difference is significant. When varying the sizes and structures of the tables, the difference is very significant; see Figure 8.

Beside comparing runtime in every query, we also compare runtime of every group presented in Figure 9. Comparing to MySQL, ADW is more than at most (6.24 times) at group 1st which uses only *Where* command, and at least (1.22 times) at group 3rd which uses *Where* and *Joint* commands.

Figure 8 Different times between MySQL and ADW in runtime of every query (see online version for colours)

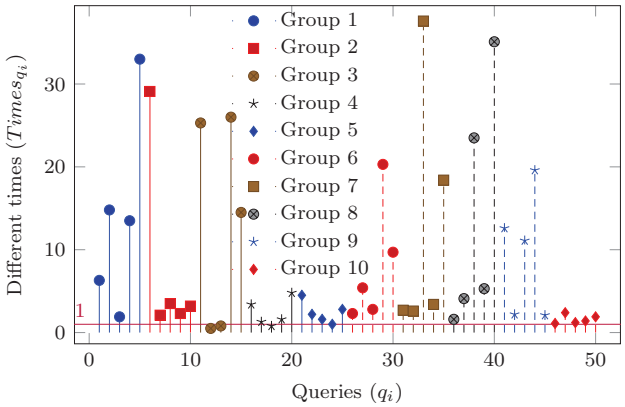


Figure 9 Different times between MySQL and ADW in runtime of every group (see online version for colours)

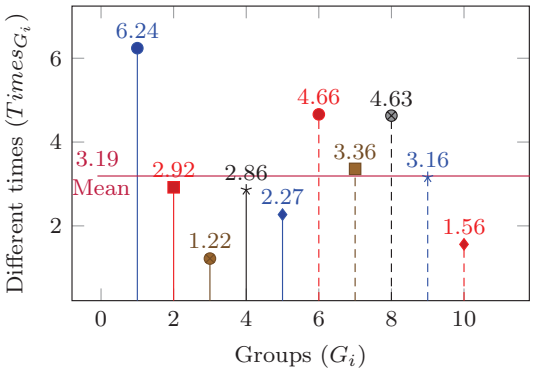


Figure 10 Average runtimes of MySQL and ADW in every groups (see online version for colours)

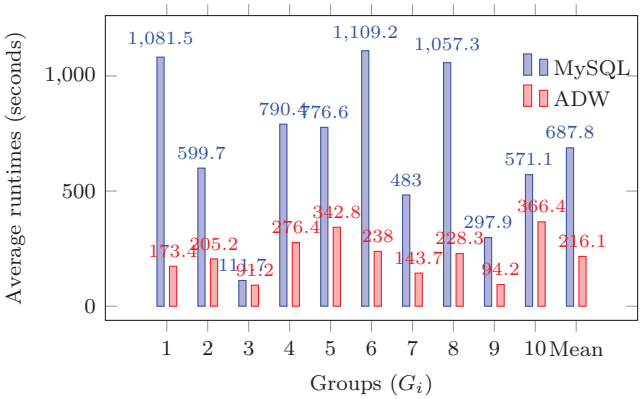


Figure 10 presents the average runtime of the 10 query groups on MySQL and ADW. Mean, the run time of a reading query on MySQL and ADW, is 687.8 seconds and 216.1 seconds, respectively. It means that ADW is faster 3.19 times. In the future, by deploying ADW solution on cloud or distributed systems, we believe that the performance will be even much better than MySQL.

9 Application for decision making

To study proposed ADW and its performance on real agricultural data, we illustrated some queries examples to show how to extract information from ADW. These queries incorporate inputs on crop, yield, pest, soil, fertiliser, inspection, farmer, businessman and operation time to reduce labour and fertiliser inputs, farmer services, disease treatment and also increase yields. This query information could not be extracted if the Origin's separate 29 datasets have not been integrated into ADW. The data integration through ADW actually improves the value of crop management data over time for better decision-making.

Example 1: List fields, crops in the fields, yield and pest in the field with conditions: (1) the fields do not used 'urea' fertiliser; (2) the crops has 'yellow rust' or 'brown rust' diseases; (3) the crops were grown in 2015.

```
select CR.CropName, FI.FieldName, FF.Yield,
       PE.CommonName, FF.PestNumber, PE.Description
from FieldFact FF, Crop CR, Field FI, Pest PE,
     Fertiliser FE, Inspection INS, OperationTime OP
where FF.CropID = CR.CropID and
      FF.FieldID = FI.FieldID and
      FF.PestID = PE.PestID and
      FF.FertiliserID = FE.FertiliserID and
      CR.CropID = INS.CropID and
      FF.OperationTimeID = OP.OperationTimeID and
      FE.FertiliserName <> 'urea' and
      (INS.Description = 'Yellow Rust' or
             INS.Description = 'Brown Rust') and
      Year(INS.Date) = '2015' and
      Year(OP.StartDate) = '2015' and
      Year(OP.EndDate) = '2015'
```

Example 2: List farmers and their crop quantities were sold by Ori Agro company in 08/2016.

```
select FA.FarmerID, FA.FarmerName, CR.CropName,
       SF.Unit, SUM(SF.Quantity)
from Salefact SF, business BU, farmer FA, crop CR
where SF.BusinessID = BU.BusinessID and
      SF.FarmerID = FA.FarmerID and
      SF.CropID = CR.CropID and
      Month(SF.SaleDate) = '08' and
      Year(SF.SaleDate) = '2016' and
      BU.BusinessName = 'Ori Agro'
group by CR.CropName
```

Example 3: List Crops and their fertiliser and treatment information. In that, crops were cultivated and harvested in 2017, Yield $>$ 10 tons/ha and attached by 'black twitch' pest. Besides, the soil in field has PH $>$ 6 and Silt $<=$ 50 mg/l.

```
Select CR.CropName, FE.FertiliserName,
       FF.FertiliserQuantity, TR.TreatmentName,
       TR.Rate, TR.TreatmentComment
From FieldFact FF, Crop CR, OperationTime OT,
     Soil SO, PEST PE, Fertiliser FE, Treatment TR
Where FF.CropID = CR.CropID and
      FF.OperationTimeID = OT.OperationTimeID and
      FF.SoildID = SO.SoilID and
      FF.PestID = PE.PestID and
      FF.FertiliserID = FE.FertiliserID and
      FF.TreatmentID = TR.TreatmentID and
      Year(OT.StartDate) = '2017' and
      Year(OT.EndDate) = '2017' and
      FF.Yield > 10 and
      SO.PH > 6 and SO.Silt <= 50 and
      PE.CommonName = 'Black twitch'
```

Example 4: List crops, fertilisers, corresponding fertiliser quantities in spring, 2017 in every field and site of 10 farmers (crop companies) who used the large amount of P_2O_5 in winter, 2016.

To execute this request, the query needs to exploit data in the FieldFact fact table and the six dimension tables, namely Crop, Field, Site, Farmer, Fertiliser and OperationTime. The query consists of two subqueries which return *10 farmers (crop companies) that used the largest amount of Urea in spring, 2016*.

```
Select FI.FieldName, SI.SiteName, FA.FarmerName,
       CR.CropName, FE.FertiliserName,
       FF.FertiliserQuantity, FE.Unit, OT.StartDate
From FieldFact FF, Crop CR, Field FI, Site SI,
     Farmer FA, Fertiliser FE, Operationtime OT
Where FF.CropID = CR.CropID and
      FF.FieldID = FI.FieldID and
      FF.FertiliserID = FE.FertiliserID and
      FF.OperationTimeID = OT.OperationTimeID and
      FI.SiteID = SI.SiteID and
      SI.FarmerID = FA.FarmerID and
      OT.Season = 'Spring' and
      YEAR(OT.StartDate) = '2017' and
      FA.FarmerID IN(
       Select FarmerID
       From
       (Select SI.FarmerID as FarmerID,
         SUM(FF.FertiliserQuantity) as SumFertiliser
        From FieldFact FF, Field FI, Site SI,
             Fertiliser FE, OperationTime OT
        Where FF.FieldID = FI.FieldID and
              FF.FertiliserID = FE.FertiliserID and
              FF.OperationTimeID =
                            OT.OperationTimeID and
              SI.SiteID = FI.SiteID and
              FE.FertiliserName = 'SO3' and
              OT.Season = 'Spring' and
              YEAR(OT.StartDate) = '2016'
         Group by SI.FarmerID
         Order by SumFertiliser DESC
         Limit 10
        )AS Table1
      )
```

10 Conclusion and future work

In this paper, we presented a schema optimised for the real agricultural datasets that were made available to us. The schema was designed as a constellation so it is flexible to adapt to other agricultural datasets and quality criteria of agricultural Big Data. Based on some existing popular open source DWs, we designed and implemented the agricultural DW by combining Hive, MongoDB and Cassandra DWs to exploit their advantages and overcome their limitations. ADW includes necessary modules to deal with large scale and efficient analytics for agricultural Big Data. Moreover, through particular reading queries using popular HQL/SQL commands, ADW storage outperforms MySQL by far. Finally, we outlined some complex HQL queries that enabled knowledge extraction from ADW to optimise the agricultural operations.

In the future work, we shall pursue the deployment of ADW on a cloud system and implement more functionalities to exploit this DW. The future developments will include:

- experimentation and analysis of the performance of MongoDB and the affectation between MongoDB and Hive
- sophisticated data mining techniques (Cai et al., 2012) to determine crop data characteristics and combined with expected outputs to extract useful knowledge
- predictive models based on machine learning algorithms
- an intelligent interface for data access
- combination with the high-performance knowledge map framework (Le-Khac et al., 2007).

Acknowledgement

This research is an extended work of Ngo et al. (2019) being part of the CONSUS research program. It is funded under the SFI Strategic Partnerships Programme (16/SPP/3296) and is co-funded by Origin Enterprises Plc. Dr. Vuong M. Ngo implemented the primary part of this research as he worked in University College Dublin, Ireland.

References

Adelman, S. and Moss, L. (2000) *Data Warehouse Project Management*, 1st ed., Addison-Wesley Professional.

Amazon document (2018) *Amazon Redshift Database Developer Guide*, Samurai ML.

Bendre, M.R., Thool, R.C. and Thool, V.R. (2015) 'Big data in precision agriculture: Weather forecasting for future farming', *International Conference on Next Generation Computing Technologies (NGCT)*, IEEE.

Cai, F., Le-Khac, N.A. and Kechadi, T. (2012) 'Clustering approaches for financial data analysis: a survey', *The 8th International Conference on Data Mining (DMIN 2012)*, pp.105–111.

Chodorow, K. (2013) *MongoDB: The Definitive Guide*, 2nd ed. (powerful and scalable data storage), O'Reilly Media.

Devitt, S.K., Perez, T., Polson, D. *et al.* (2017) 'A cognitive decision tool to optimise integrated weed management', *Proceedings of International Tri-Conference for Precision Agriculture*.

Dicks, L.V., Walsh, J.C. and Sutherland, W.J. (2014) 'Organising evidence for environmental management decisions:a '4' hierarchy', *Trends in Ecology and Evolution*, Vol. 29, No. 11, pp.607–613.

Du, D. (2018) *Apache Hive Essentials*, 2nd ed., Packt Publishing.

EC report (2016) *Europeans, Agriculture and the Common Agricultural Policy*, Special Eurobarometer 440, The European Commission.

FAO-CSDB report (2018) *Global Cereal Production and Inventories to Decline but Overall Supplies Remain Adequate*, Release date: 6 December, 2018, Cereal Supply and Demand Brief, FAO.

FAO-FSIN report (2018) *Global Report on Food Crises 2018*, Food Security Information Network, FAO.

Friedman, S.P., Communar, G. and Gamliel, A. (2016) 'Didas – user-friendly software package for assisting drip irrigation design and scheduling', *Computers and Electronics in Agriculture*, Vol. 120, pp.36–52.

Golfarelli, M. and Rizzi, S. (2009) *Data Warehouse Design: Modern Principles and Methodologies*, McGraw-Hill Education.

Gupta, A., Yang, F., Govig, J. *et al.* (2016) 'Mesa: a geo-replicated online data warehouse for google's advertising system', *Communications of the ACM*, Vol. 59, No. 7, pp.117–125.

Gutierrez, F., Htun, N.N. *et al.* (2019) 'A review of visualisations in agricultural decision support systems: An HCI perspective', *Computers and Electronics in Agriculture*.

Hafezalkotob, A., Hami-Dindar, A., Rabie, N. and Hafezalkotob, A. (2018) 'A decision support system for agricultural machines and equipment selection: a case study on olive harvester machines', *Computers and Electronics in Agriculture*, Vol. 148, pp.207–216.

Han, E., Ines, A.V.M. and Baethgen, W.E. (2017) 'Climate-agriculture-modeling and decision tool (camdt): a software framework for climate risk management in agriculture', *Environmental Modelling and Software*, Vol. 95, pp.102–114.

Hewitt, E. and Carpenter, J. (2016) *Cassandra: The Definitive Guide*, 2nd ed. (distributed data at web scale), O'Reilly Media.

Hows, D. and et al. (2015) *The Definitive Guide to MongoDB*, 3rd ed. (a complete guide to dealing with big data using MongoDB), Apress.

Huang, Y., Sui, R., Thomson, S.J. and Fisher, D.K. (2013) 'Estimation of cotton yield with varied irrigation and nitrogen treatments using aerial multispectral imagery', *International Journal of Agricultural and Biological Engineering*, Vol. 6, No. 2, pp.37–41.

Inmon, W.H. (2005) *Building the Data Warehouse*, Wiley.

Kamilaris, A., Assumpcio, A., Blasi, A.B., Torrellas, M. and Prenafeta-Boldú, F.X. (2018) *Estimating the Environmental Impact of Agriculture by Means of Geospatial and Big Data Analysis: The Case of Catalonia*, Springer, pp.39–48.

Kimball, R. and Ross, M. (2013) *The data warehouse toolkit: the definitive guide to dimensional modeling (3rd edition)*, Wiley.

Lam, C.P. and et al. (2016) *Hadoop in Action*, 2nd ed., Manning.

Le-Khac, N.A., Aouad, L.M. and Kechadi, T. (2007) 'Distributed knowledge map for mining data on grid platforms', *Int. J. Computer Science and Network Security*, Vol. 7, No. 10, pp.98–107.

Lokers, R., Knapen, R., Janssen, S., Randen, Y. and Jasen, J. (2016) 'Analysis of big data technologies for use in agro-environmental science', *Environmental Modelling and Software*, Vol. 48, pp.494–504.

Lundstrom, C. and Lindblom, J. (2018) 'Considering farmers' situated knowledge of using agricultural decision support systems (agridss) to foster farming practices: the case of cropsat', *Agricultural Systems*, Vol. 159, pp.9–20.

Neeraj, N. (2015) *Mastering Apache Cassandra*, 2nd ed., Packt Publishing.

Ngo, V.M., Le-Khac, N.A. and Kechadi, M.T. (2018) 'An efficient data warehouse for crop yield prediction', *The 14th International Conference Precision Agriculture (ICPA-2018)*, pp.3:1–3:12.

Ngo, V.M., Le-Khac, N.A. and Kechadi, M.T. (2019) 'Designing and implementing data warehouse for agricultural big data', *The 8th International Congress on BigData (BigData-2019)*, Springer-LNCS, Vol. 11514, pp.1–17.

Ngo, V.M. and Kechadi, M.T. (2020) 'Crop knowledge discovery based on agricultural big data integration', *The 4th International Conference on Machine Learning and Soft Computing (ICMLSC)*, ACM, pp.1–5.

Nilakanta, S., Scheibe, K.P. and Rai, A. (2008) 'Dimensional issues in agricultural data warehouse designs', *Computers and Electronics in Agriculture*, Vol. 60, No. 2, pp.263–278.

Oliver, D.M., Bartie, P.J., Heathwaite, A.L., Pschetz, L. and Quilliam, R.S. (2017) 'Design of a decision support tool for visualising e. coli risk on agricultural land using a stakeholder-driven approach', *Land Use Policy*, Vol. 66, pp.227–234.

Oracle document (2017) *Database Data Warehousing Guide*, Oracle12c doc release 1.

Origin report (2018) *Annual Report and Accounts*, Origin Enterprises plc.

Pantazi, X.E. (2016) 'Wheat yield prediction using machine learning and advanced sensing techniques', *Computers and Electronics in Agriculture*, Vol. 121, pp.57–65.

Paredes, P., Rodrigues, G.C., Alves, I. and Pereira, L.S. (2014) 'Partitioning evapotranspiration, yield prediction and economic returns of maize under various irrigation management strategies', *Agricultural Water Management*, Vol. 135, pp.27–39.

Park, S., Im, J., Jang, E. and Rhee, J. (2016) 'Drought assessment and monitoring through blending of multi-sensor indices using machine learning approaches for different climate regions', *Agricultural and Forest Meteorology*, Vol. 216, pp.157–169.

Protopop, I. and Shanoyan, A. (2016) 'Big data and smallholder farmers: Big data applications in the agri-food supply chain in developing countries', *International Food and Agribusiness Management Review, IFAMA*, Vol. 19, No. A, pp.1–18.

Rembold, F., Meroni, M., Urbano, F. *et al.* (2019) 'Asap: A new global early warning system to detect anomaly hot spots of agricultural production for food security analysis', *Agricultural Systems*, Vol. 168, pp.247–257.

Rogovska, N., Laird, D.A., Chiou, C.P. and Bond, L.J. (2019) 'Development of field mobile soil nitrate sensor technology to facilitate precision fertilizer management', *Precision Agriculture*, Vol. 20, No. 1, pp.40–55.

Ruan, J., Wang, Y., Chan, F.T.S. *et al.* (2019) 'A life cycle framework of green iot-based agriculture and its finance, operation, and management issues', *IEEE Communications Magazine*, Vol. 57, No. 3, pp.90–96.

Rupnik, R., Kukar, M., Vracar, P. and Kosir, D. (2019) 'Agrodss: a decision support system for agriculture and farming', *Computers and Electronics in Agriculture*, Vol. 161, pp.260–271.

Schnase, J.L., Duffy, D., Tamkin, G.S. *et al.* (2017) 'Merra analytic services: meeting the big data challenges of climate science through cloud-enabled climate analytics-as-a-service', *Computers, Environment and Urban Systems*, Vol. 161, pp.198–211.

Schuetz, C.G., Schausberger, S. and Schrefl, M. (2018) 'Building an active semantic data warehouse for precision dairy farming', *Organizational Computing and Electronic Commerce*, Vol. 28, No. 2, pp.122–141.

Schulze, C., Spilke, J. and Lehner, W. (2007) 'Data modelling for precision dairy farming within the competitive field of operational and analytical tasks', *Computers and Electronics in Agriculture*, Vol. 59, Nos. 1–2, pp.39–55.

Udias, A., Pastori, M., Dondeynaz, C. *et al.* (2018) 'A decision support tool to enhance agricultural growth in the mÃ©krou river basin (West Africa)', *Computers and Electronics in Agriculture*, Vol. 154, pp.467–481.

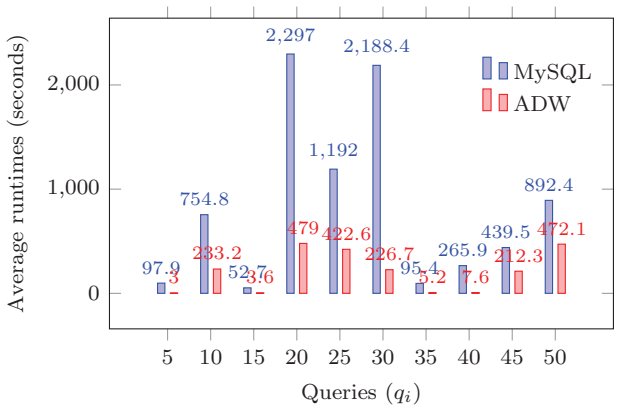
UN document (2017) *World Population Projected to Reach 9.8 billion in 2050, and 11.2 billion in 2100*, Department of Economic and Social Affairs, United Nations.

USDA report (2018) *World Agricultural Supply and Demand Estimates 08/2018*, United States Department of Agriculture.

Appendix

The followings are HQL/SQL scripts of 10 queries which are representative of 10 query groups. The average runtimes of these queries on MySQL and ADW are shown in Figure A1.

A1 Average runtimes of MySQL and ADW in 10 typical queries (see online version for colours)



1) The query 5th belongs to the group 1st:

```
SELECT fieldfact.FieldID, crop.cropname,
       fieldfact.yield
FROM fieldfact, crop
WHERE fieldfact.cropid = crop.cropid and
      SprayQuantity = 7 and
      (crop.CropName like 'P\%' or
       crop.CropName like 'R\%' or
       crop.CropName like 'G\%');
```

2) The query 10th belongs to the group 2nd:

```
SELECT soil.PH, count(*)
FROM fieldfact, soil
WHERE fieldfact.SoildID = soil.SoilID and
      fieldfact.sprayquantity = 2
GROUP by soil.PH;
```

3) The query 15th belongs to the group 3rd:

```
SELECT fieldfact.yield,
       fertiliser.fertiliserName,
       fertiliser.fertiliserGroupName
FROM fieldfact
RIGHT JOIN fertiliser on
     fieldfact.fertiliserID = fertiliser.fertiliserID
WHERE fieldfact.fertiliserQuantity = 10 and
      fertiliser.fertiliserName like '%slurry%';
```

4) The query 20th belongs to the group 4th:

```
SELECT sprayproductname
FROM fieldfact, spray
WHERE fieldfact.sprayid = spray.sprayid and
      fieldfact.watervolumn > 5 and
      fieldfact.watervolumn < 20
UNION
SELECT productname
FROM product, orderfact
WHERE product.ProductID = orderfact.ProductID
      and (orderfact.Quantity = 5 or
           orderfact.Quantity = 6);
```

5) The query 25th belongs to the group 5th:

```
SELECT fieldfact.fieldID, field.FieldName,
       field.FieldGPS, spray.SprayProductName
FROM fieldfact, field, spray
WHERE fieldfact.FieldID = field.FieldID and
      fieldfact.SprayID = spray.SprayID and
      fieldfact.PestNumber = 6
ORDER BY field.FieldName;
```

6) The query 30th belongs to the group 6th:

```
SELECT fieldfact.FieldID, nutrient.NutrientName,
       nutrient.Quantity, nutrient.`Year`
FROM fieldfact
RIGHT JOIN nutrient on
     fieldfact.NutrientID = nutrient.NutrientID
WHERE fieldfact.NutrientQuantity = 3 and
      fieldfact.fertiliserquantity = 3
ORDER BY nutrient.NutrientName
LIMIT 10000;
```

7) The query 35th belongs to the group 7th:

```
SELECT crop.cropname,
       sum(fieldfact.watervolumn) as sum1
FROM fieldfact, crop
WHERE fieldfact.cropid = crop.cropid and
      fieldfact.sprayquantity = 8 and
      crop.EstYield >= 1 and crop.EstYield <=10
GROUP BY crop.cropname
HAVING sum1 > 100;
```

8) The query 40th belongs to the group 8th:

```
SELECT crop.cropname,
       sum(fieldfact.fertiliserquantity) as sum1
FROM fieldfact, crop
WHERE fieldfact.cropid = crop.cropid and
      fieldfact.nutrientquantity= 5 and
      crop.EstYield <=1
GROUP by crop.cropname
HAVING sum1 > 30
ORDER BY crop.cropname;
```

9) The query 45th belongs to the group 9th:

```
SELECT nutrient.NutrientName,
       sum(nutrient.Quantity) as sum1
FROM fieldfact
LEFT JOIN nutrient on
    fieldfact.NutrientID = nutrient.NutrientID
WHERE nutrient.nutrientName like '%tr%' and
     (fieldfact.pestnumber = 16 or
      fieldfact.pestnumber = 15)
GROUP by nutrient.NutrientName
HAVING sum1 <300
ORDER BY nutrient.NutrientName;
```

10) The query 50th belongs to the group 10th:

```
SELECT sprayproductname as name1,
       sum(fieldfact.watervolumn) as sum1
FROM fieldfact, spray
WHERE fieldfact.sprayid = spray.sprayid and
      fieldfact.Yield > 4 and fieldfact.Yield < 8
GROUP by sprayproductname
HAVING sum1 > 210
UNION
SELECT productname as name1,
       sum(orderfact.Quantity) as sum2
FROM product, orderfact
WHERE product.ProductID = orderfact.ProductID and
     (orderfact.Quantity = 5 or
      orderfact.Quantity = 6)
GROUP by productname
HAVING sum2 > 50
ORDER BY name1;
```