FIMaaS: Scalable Frequent Itemset Mining-as-a-Service on Cloud for Non-Expert Miners

Zhao Han
Department of Computer Science
University of Manitoba
Winnipeg, MB
Canada
umhan35@cs.umanitoba.ca

Carson K. Leung*
Department of Computer Science
University of Manitoba
Winnipeg, MB
Canada
kleung@cs.umanitoba.ca

ABSTRACT

Frequent itemset mining discovers implicit, previously unknown and potentially useful knowledge—in the form of frequent itemsets—from data. For example, discovery of frequently purchased merchandise products reveals customer purchase patterns, which help store managers about their business strategies and promotional tactics. These, in turn, help increase profits of the stores. As another example, discovery of popular collections of courses reveals course popularity and trends of some subject matters. These, in turn, assist university administrators schedule courses and their corresponding exams to avoid conflict or exam hardship, as well as planning of the calendar. As we are living in the era of big data, many applications and services generate high volumes of a wide variety of highly valuable data at a high velocity. These data can be of a wide range of veracity. Consequently, having scalable frequent itemset mining service is important to both the data mining experts and non-experts. Over the past two decades, numerous frequent itemset mining algorithms have been proposed. Many of them require some degrees of data mining knowledge and expertise, which may be inaccessible by layman. In this paper, we propose a tool with an intention to provide scalable *frequent itemset mining-as-a-service* (*FIMaaS*) on cloud for non-expert data miners.

CCS Concepts

- **Information systems → Information systems applications → Data mining → Association rules;**
- **Human-centered computing → Visualization → Visualization techniques;**
- **Human-centered computing → Visualization → Visualization application domains → Visual analytics;**
- **Computer systems organization → Architectures → Distributed architectures → Cloud computing;**

Keywords

Application-as-a-service (AssS), data mining, frequent patterns

1. INTRODUCTION

Most of us are eager for knowledge because knowledge empowers us and enriches our life. We used to gain knowledge through our experience and perception of the world over the last few thousand years, through education. In the current information age, computers allow us to collect and store a large amount of data. Due to the high volume of data, one may not be able to easily transform all those data into valuable knowledge or information.

Data mining [5] becomes a necessity to help us discover knowledge. Specifically, it refers to the discovery of implicit, previously unknown and potentially useful knowledge. As a fundamental data mining tasks, *frequent itemset mining* (*FIM*) [1] finds one kind of knowledge called frequently occurring patterns in the form of *sets of items* (shorthanded as *itemsets*). In other words, frequent itemset mining discovers implicit, previously unknown and potentially useful knowledge—in the form of frequent patterns—from data. For example, discovery of frequently purchased merchandise products reveals customer purchase patterns, which help store managers about their business strategies and promotional tactics. These, in turn, help increase profits of the stores. As another example, discovery of popular collections of courses reveals course popularity and trends of some subject matters. These, in turn, assist university administrators schedule courses and their corresponding exams to avoid conflict or exam hardship, as well as planning of the calendar.

Since the introduction of frequent pattern mining [2], researchers have been focused on algorithmic efficiency but mostly ignoring the interactiveness with users. Many of these algorithms are efficient in terms of memory or disk space, CPU cycles, and/or data characteristics (e.g., dense vs. sparse data). However, they mostly do not provide user-

*Corresponding author: C.K. Leung

BigDAS'15, Oct. 20–23, 2015, Jeju Island, Republic of Korea (S. Korea)

ISBN 978-1-4503-3846-2/15/10…$15.00

DOI: http://dx.doi.org/10.1145/2837060.2837072

friendly interfaces for non-expert miners to better understand mined information. While frequent patterns mined are immediately useful for non-experts (e.g., store managers, university administrators), frequent pattern mining process in particular (and the boarder data mining process in general) often requires some levels of expertise, which may be inaccessible to layman. Without friendly-user interfaces, some potential users of data mining algorithms or systems may find these algorithms or systems challenging to use. This, in turn, generates a negative effect of discouraging users to analyze and explore the data and to discover some valuable knowledge or information embedded in the data.

Due to advances in technology, high volumes of valuable data (e.g., streams of banking, financial, and shopper market basket data) are collected or generated at a high velocity in high varieties of data sources in various real-life business, engineering, and scientific applications and services in modern organizations and society. Due to their high volumes, the quality and accuracy of these data depend on their veracity (uncertainty of data). This leads us into the new era of *big data* [10, 8]. Embedded in these big data is implicit, previously unknown, and potentially useful information and knowledge. However, these big data come with volumes beyond the ability of commonly-used software to capture, manage, and process within a tolerable elapsed time.

In general, characteristics of these big data can be described by the well-known 5V's:

1. *volume*, which focuses on the quantity of data;
2. *value*, which focuses on the usefulness of data (e.g., knowledge that can be discovered from the big data);
3. *velocity*, which focuses on the speed at which data are collected or generated;
4. *variety*, which focuses on differences in types, contents, or formats of data; and
5. *veracity*, which focuses on the quality of data (e.g., precise data, uncertain and imprecise data).

Hence, new forms of information science and technology—such as scalable big data analytics and mining algorithms, data science systems, as well as business intelligence (BI) solutions—are needed to process and analyze these big data so to as enable enhanced decision making, insight, knowledge discovery, and process optimization. For instance, *cloud computing* offers dynamic scalability through virtualization, which is capable of handling large quantity of transactional data. To get mining results faster, users have to invest on their own infrastructure, which small and medium business owners may not be able to afford or may not have the expertise to operate such an infrastructure. On the other hand, due to the prevalence of Internet used in cloud computing, a cloud-enabled web interface allows the service accessible from everywhere, on any types of devices, and by various kinds of users with different levels of expertise. Practitioners who develop applications or services can also leverage existing cloud solutions (e.g., Amazon Web Service) to ease the development effort. Last but not least, the interface also hides the complexity of data mining algorithms in general—and frequent itemset mining algorithms in particular—and removes the burden on choosing an algorithm most suitable to the data (which may be dense or sparse) at hand.

In this paper, we propose a cloud-based web application solution—called *FIMaaS*—to provide non-expert data miners with *frequent itemset mining-as-a-service* (*FIMaaS*) on cloud. This highly scalable and easily accessible service for finding frequent patterns is *our key contribution* of this paper. To the best of our knowledge, this is the first frequent itemset mining system where miners can simply upload data files and get frequent patterns without worrying if the machine is capable of processing my data. The size of an uploaded data file is currently set to 5 GB maximum because of the cost of cloud compute power. However, this limitation can be lifted and the design of our system is capable of handling petabytes (or more) of uploaded data if needed. In order to get frequent patterns quickly, FIMaaS is backed by the following:

- *Amazon Simple Storage Service* (*S3*)[1], which is an unlimited storage service; and
- *Amazon Elastic MapReduce* (*EMR*)[2], which is an on-demand cluster compute service to process large data.

In addition, *Ruby on Rails*[3] is used for web frontend as it allows rapid development of an easy-to-use web interface and user management system.

Note that, due to the abstraction through virtualization of physical IT resources offered by cloud computing, the service delivered to customers can be classified into three layers:

1. *infrastructure-as-a-service* (*IaaS*), which offers IT resources;
2. *platform-as-a-service* (*PaaS*), which provides platforms for applications; and
3. *software-as-a-service* (*SaaS*), which is visible and directly interacts with end users.

Amazon S3 and *Amazon Elastic Compute Cloud* (*EC2*)[4] are examples of IaaS as (a) Amazon S3 provides storage resources and (b) Amazon EC2 provides processing power. Ruby on Rails is an example of PaaP as it offers managed runtime management for applications to ease deployment and scalability effort. Our FIMaaS is an example of SaaS.

The remainder of this paper is organized as follows. The next section presents related works. Section 3 first introduces our research problem of providing data mining-as-a-service (DMaaS), and then describes the design and architecture of our proposed FIMaaS system in detail. Evaluation and conclusions are given in Sections 4 and 5, respectively.

2. RELATED WORKS

Delivering data mining as a service, which attempts to lower barriers to data mining tasks is not a new concept. For instance, Sarawagi and Nagaralu [13] provided *data mining-as-a-service* (*DMaaS*) over the Internet. Zorrilla et al. [16] proposed a data mining web interface for non-expert users in an education setting. The web interface is user friendly, but it does not take advantage of cloud computing (e.g., scalability and virtualization the cloud has to offer). The quantity

[1]https://aws.amazon.com/s3/
[2]https://aws.amazon.com/elasticmapreduce/
[3]http://rubyonrails.org/
[4]https://aws.amazon.com/ec2/

of the data mined was also not large because the system does not scale well. Ramya et al. [12] developed a knowledge extraction system for non-expert miners. However, their system only generates a trivial report with simple graphs (cf. data mining puts emphasis on extraction of implicit and previously unknown information). Guedes et al. [6] developed Anteater, which is a service-oriented architecture for data mining. Anteater relies on web services to offer a simple interface to users and support computationally intensive processing through parallelism. Their experiments show that (a) Anteater is 16 times faster than non-distributed systems and (b) novice users understand the mined results without further data mining knowledge. In other words, Anteater does not show the mined frequent patterns. In contrast, we focus on frequent itemset mining and leverages cloud computing to achieve the goal.

Because (a) frequent itemset mining (FIM) is a generic field and (b) frequent patterns can be found in a variety of data, the *applications and services of frequent itemset mining* are of broad range and keep springing up. This also leads to potentially wide use of FIMaaS, which serves as one of the motivations. In 1994, the research problem of the frequent itemset mining [2] was introduced in the context of an application of analyzing shopper basket market data to mine frequent itemsets and form interesting association rules that reveal customer buying behaviors. Chen et al. [4] applied frequent itemset mining to web logs to find frequent access patterns, which can help to maximize visitors' accesses [4]. Mining of frequent access patterns also help to differentiate students' access behaviors for online learning [15]. Moreover, frequent access patterns can also be used for personalization, which recommends web pages [11].

In addition, frequent itemset mining can also be applied to time series data. Han et al. [7] found periodic patterns in temporal data, while Bettini et al. [3] used FIM to detect frequent event patterns.

Furthermore, researchers also applied FIM to other data mining tasks (e.g., clustering). For instance, Wang et al. [14] proposed a clustering algorithm for transactional database by finding frequent overlapping partial transactions. Note that the aforementioned frequent itemset mining algorithms tend to be developed for some specific domains. As a result, the resulted achievements are not immediate available to non-expert miners.

3. FIMaaS: FREQUENT ITEMSET MINING-AS-A-SERVICE

To build a system aimed for non-expert miners, we implement a scalable frequent itemset mining system with easily accessible and user-friendly interface. With this system, those who do not have FIM expertise want to have the FIM job done fast and on demand without knowing how a potentially large dataset is handled and what the underlying algorithm is and how to choose an efficient one.

To meet the needs of non-experts, the system can be divided into the following sub-problems:

1. input,
2. processing, and
3. output.

First, the system needs to handle large input (i.e., data file). Users can expect fast and reliable upload speed while secure transmission and storage are guaranteed. While users are waiting for output (i.e., frequent patterns), they would like to obtain real-time feedback of the status of the job. When the job is done, frequent patterns can be retrieved easily. In all sub-problems, users can also expect high availability and high scalability of the system, which means many users can synchronously upload data files and get results quickly.

3.1 Architecture

When designing our FIMaaS system, we focus on the following two aspects:

1. *"cloud-enablility"*, which makes the service user friendly for non-expert users in terms of easiness of use and the total mining time; and
2. *scalability*, which is offered by cloud computing.

In order to make the service easily scalable, service-oriented computing paradigm is adopted. Specifically, our FIMaaS system is composed of several services, and each part can work independently. Note that the total mining time is directly related to the scalability of the service.

Figure 1 shows architecture of our FIMaaS system. Here, *users* register and sign in to FIMaaS website and can directly upload datasets to the *Amazon Simple Storage Service* (*S3*) file store through FIMaaS. In the interim, S3 notifies FIMaaS with the upload progress and when it is finished. After a dataset is finished uploading onto S3, our FIMaaS system creates a job in a *queue*. The *FIMaaS worker* then actively polls jobs from the queue and leverages the *Amazon Elastic MapReduce* (*EMR*) computer cluster to process the datasets (i.e., to mine frequent patterns from the datasets). Afterwards, The FIMaaS worker puts the output file containing the mined frequent patterns back onto S3 and informs FIMaaS. Finally, FIMaaS returns the mined frequent patterns to users, who also have the option to download the output file containing the mined frequent patterns.

Note that the Amazon S3 file store—which is offered by *Amazon Web Service* (*AWS*)[5]—provides fast, reliable and safe transmission, unlimited storage, and scalability. By leveraging S3, a user can upload a dataset file as fast as the user's Internet connection. In the current era of big data, it is not uncommon to have datasets ranging from gigabytes to terabytes in size. Hence, S3 can satisfy this requirement of data volumes. Currently, FIMaaS allows users to upload datasets as large as 5 TB. In FIMaaS, data transmission is offloaded to S3. Data are directly transferred to S3 through user's browsers without interacting with the web application server of FIMaaS. The web interface for upload is here to generate and collect input parameters (e.g., data file path and authentication information, which are needed to upload to S3). By doing this, the web application server of FIMaaS is not blocked by the file uploading process, and thus is not forcing other users to wait. Thanks to S3, FIMaaS is then empowered to handle as many concurrent uploads as requested.

Data are transmitted to S3 through a secure sockets layer (SSL)-encrypted application programming interface (API) endpoints using secure hypertext transfer protocol (HTTPS), which ensures the *security* and *integrity* of the transfer. The S3 SSL-encrypted API endpoint is temporally authorized from S3 per upload request through the FIMaaS

[5]https://aws.amazon.com/

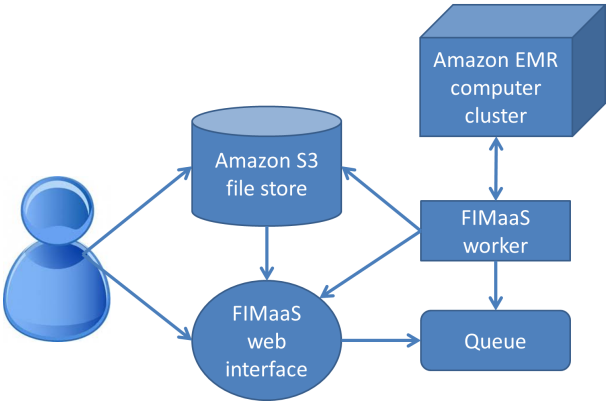


Figure 1: Architecture of our FIMaaS system

web interface. It prevents malicious users to abuse S3 resource without being authorized first. During the transfer process, data cannot be retrieved without the encrypted key stored on S3 servers. In the meantime, data cannot be modified or tampered without notice on the client side because each fragment of data send through SSL has an authentication code for the receiver to authenticate when the next fragment of data are ready to transmit.

After users upload data to S3, a job is queued (cf. immediately processed). In other words, the data processing is asynchronous. This increases the throughput of the FIMaaS web interface, and thus making FIMaaS scalable. FIMaaS—which is an application that is constantly checking the job queue through API of FIMaaS—consumes the jobs in the queue if any.

FIMaaS worker passes the jobs with necessary parameters to a cluster computing system to do the heavy lifting. When the computer cluster is mining for frequent patterns, the user is getting notification about the mining process (e.g., "being processed") on the web interface. Within the FIMaaS worker, the job processing component—which is implemented in Java—is responsible for taking jobs off the queue and changing the status of a job (e.g., "processed and available for download"). The FIMaaS worker itself—which is built on top of Apache Spark[6]—is a general cluster computing system specialized in processing big data. The Machine Learning Library (MLlib)[7] in Apache Spark helps to mine frequent patterns from data. The built-in algorithm in MLlib is Parallel FP-growth (PFP) [9], which partitions user data across the compute nodes of a computer cluster to process the data in parallel.

The computer cluster used by the FIMaaS worker is hosted on Amazon EMR for easy cluster management. With EMR, our FIMaaS system leverages its cluster management provide by the Apache Hadoop[8] Yet Another Resource Negotiator (YARN) to allocate and manage compute resources on master and slave compute nodes. EMR also offers tight integration with the Amazon S3 file store. Because EMR servers and S3 servers are in the same data center, it helps decrease communication cost and maximize transfer speed between them.

[6]https://spark.apache.org/
[7]https://spark.apache.org/mllib/
[8]http://hadoop.apache.org/

Besides the job processing component (which takes jobs off the queue and changes the status of a job), another component within the FIMaaS worker is the data processing component. This component takes the following two input parameters via a simple interface:

1. Uniform Resource Identifier (URI) of the data file on S3, and
2. minimum count of a frequent product or product set (i.e., user-specified minimum frequency/support *min-sup* threshold).

When mining frequent patterns, the data processing component within the FIMaaS worker stores it back to S3. The user waiting on the web page is then given options to either visualize frequent patterns or download the file. The download link is temporary and expires within a specific period of time to ensure nobody else can download users' data.

To achieve scalability, all components of our FIMaaS system can be run independent to each other, and hence can be substituted for higher performance if needed. For example, the Amazon S3 file store can be replaced by any file store that is compatible with S3 API such as DreamObjects[9]. The FIMaaS worker, along with the Amazon EMR computer cluster, can be replaced by any frequent pattern mining algorithm that (a) can check, remove, and change the status of tasks from the job queue through FIMaaS API, (b) uses a consistent data format as required by the FIMaaS worker, and (c) puts the mined frequent patterns onto S3. Moreover, MLlib in Apache Spark (e.g., PFP algorithm) can also be replaced by any frequent pattern algorithm that requires the same input data format and output data format.

3.2 Web interface

The user-friendly interface of our FIMaaS system is a web application hosted on cloud—specifically, Amazon Elastic Compute Cloud (EC2). By doing so, the web application can be accessed everywhere, and users can use it to mine frequent patterns wherever the data is located. For example, users can choose and upload data stored in their Dropbox accounts wherever there is an Internet connection. Amazon EC2 is chosen because it scales well when the number of web application users and uploads increases. Due to the

[9]https://www.dreamhost.com/cloud/storage/

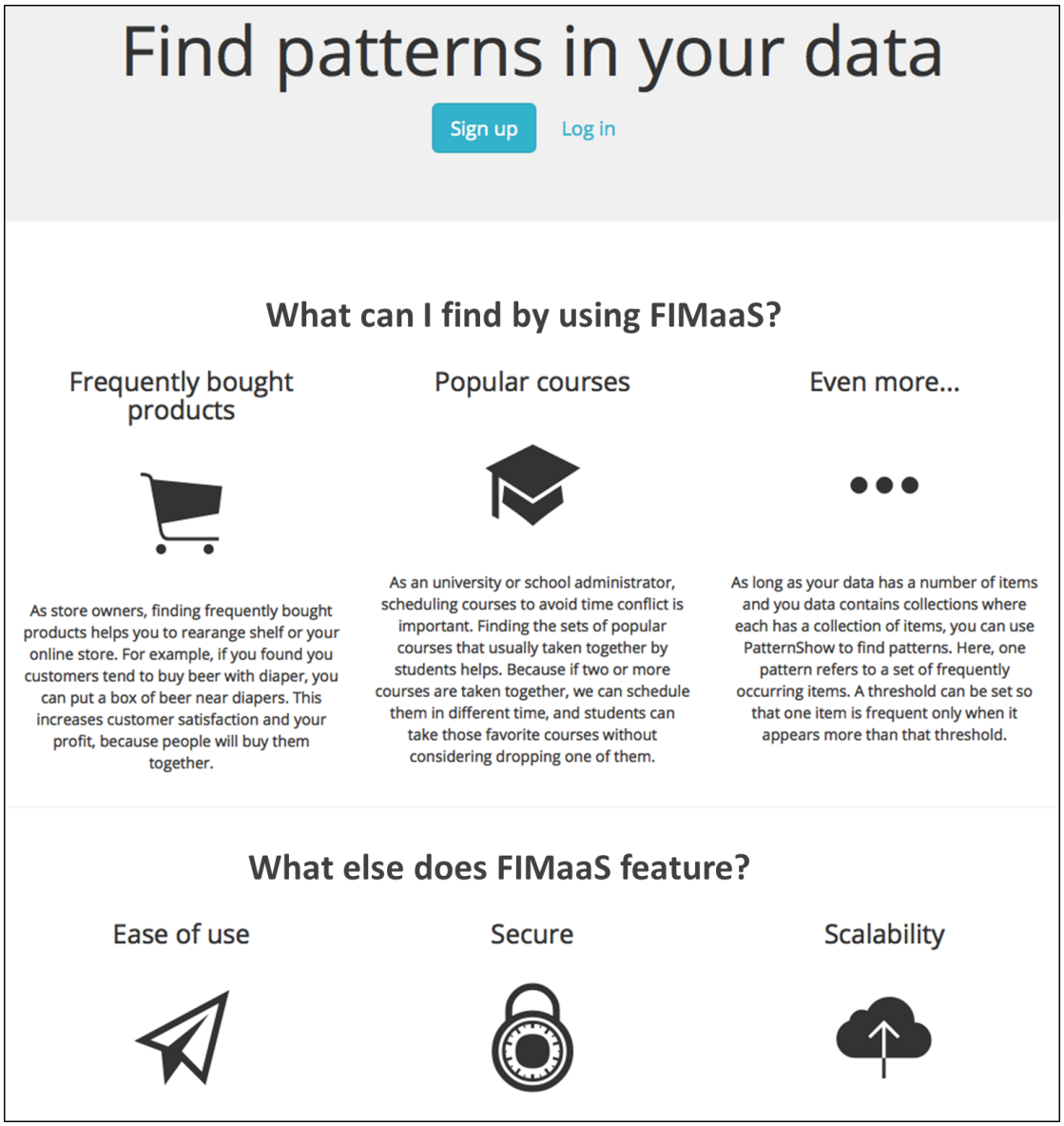


Figure 2: Homepage of our FIMaaS system

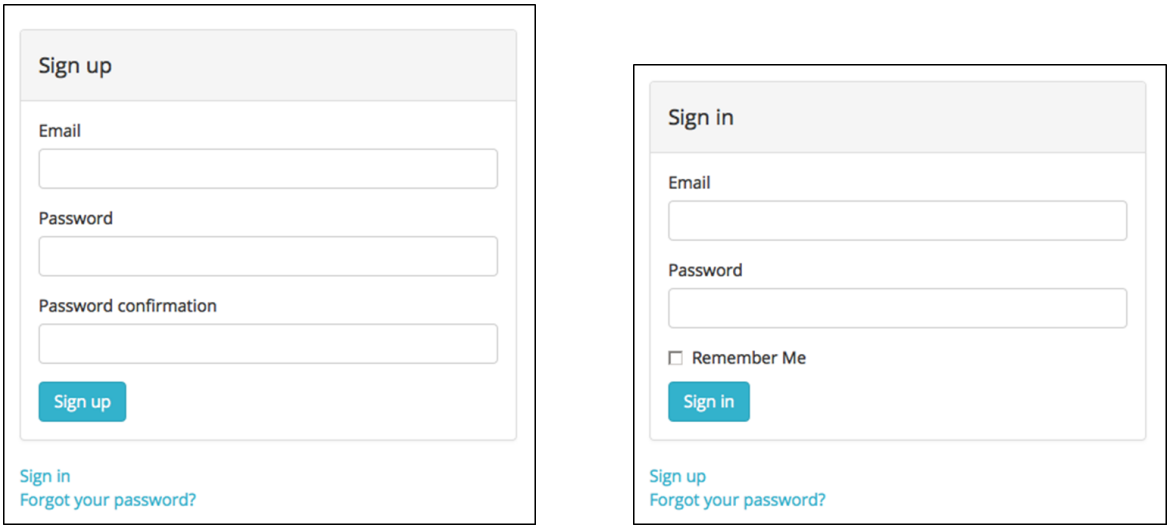
virtualization technique used, more EC2 servers can be dynamically added when the load of application increases. At the same time, those additional servers can be removed when the load decreases. The web application allows users easily register for FIMaaS and sign in, upload their transaction datasets, and obtain the frequent patterns mined from their data. User registration allows (a) easy management and access control of data for each user, as well as (b) easy retrieval when data is processed.

Figure 2 shows the homepage of our FIMaaS system. In the wide gray area on this homepage, users can click either one of the following buttons:

- Sign up button, which allows users to register an account. Figure 3(a) shows the sign-up page, in which users just need to type email and choose a password for subsequent log in.
- Log in button, which allows users to sign into FIMaaS. Figure 3(b) shows the login page, in which users type the email and password to sign in. On the login/ sign-in page, users can check the box for "Remember Me" so that they do not need to retype the email and password for later visits. If users forgot their password, they can reset the password by clicking the link "Forgot your password?" on the bottom of the page.

After users successfully logged in, they will be redirected to a page to upload their data. By clicking the "See all my data" button, users can see all of their previously uploaded data. See Figure 4. Once the user data are uploaded, users can get the data processed by setting the minimum support $minsup \in (0.0, 1.0]$.

Because the web application is responsible for user man-



(a) Sign-up page (b) Sign-in page (i.e., log-in page)

Figure 3: Sign-up and log-in pages of our FIMaaS system

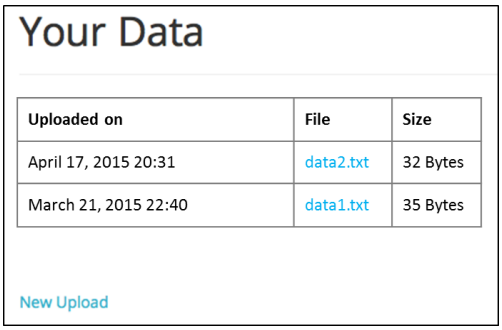


Figure 4: FIMaaS shows information about the uploaded data

agement and providing user interface, it does not require much compute power. Indeed, the web server used during development is a t2.micro instance[10] provided by Amazon EC2: it has 1 GB memory and runs on a 0.25 GHz CPU with the capability to burst to 2.5 GHz for 6 minutes per hour. As (a) data transfer is offloaded to Amazon S3 file store without interacting with the web application and (b) frequent pattern mining is offloaded to Amazon EMR computer cluster, the web application—including the web server software—only consumes half of memory (450 MB) and a small fraction of CPU cycles (say, 2%, 0.05 GHz) when idle. During the load test, it can handle 300 requests per second, which is sufficiently good as a starting point.

4. EVALUATION

While we faced difficulties and challenges during implementation, the results are encouraging and show that FIMaaS really eases the effort to find frequent patterns for non-experts and can also be very useful for expert miners. Recall from Figure 2, our FIMaaS system allows non-expert users who in different domains to analyze data and discover implicit, previously unknown and potentially useful knowledge from the data. Examples include the following:

- Store owners (as one type of non-expert users) can use FIMaaS to discover *frequently bought products*. The discovered customer behavior helps store owners to rearrange shelves or online stores so as to satisfy customers and increase the store profile; it also helps store owners in inventory and effective product promotion.

- University or school administrators (as another type of non-expert users) can use FIMaaS to discover *popular courses*. The discovered knowledge helps university or school administrators to schedule courses so as to avoid time conflict or exam hardship, to offer these popular courses more frequently, and to recommend these popular courses to new students.

4.1 Scalability

FIMaaS was tested with various transactional datasets available from Frequent Itemset Mining Dataset Repository[11]. The sizes of all datasets range from a few bytes to 1.48 GB, with the latter dataset containing almost 1.7 million transactions about 5.3 million distinct domain items. Frequent patterns are successfully found in all the

[10]https://aws.amazon.com/ec2/instance-types/#general-purpose

[11]http://fimi.ua.ac.be/data/

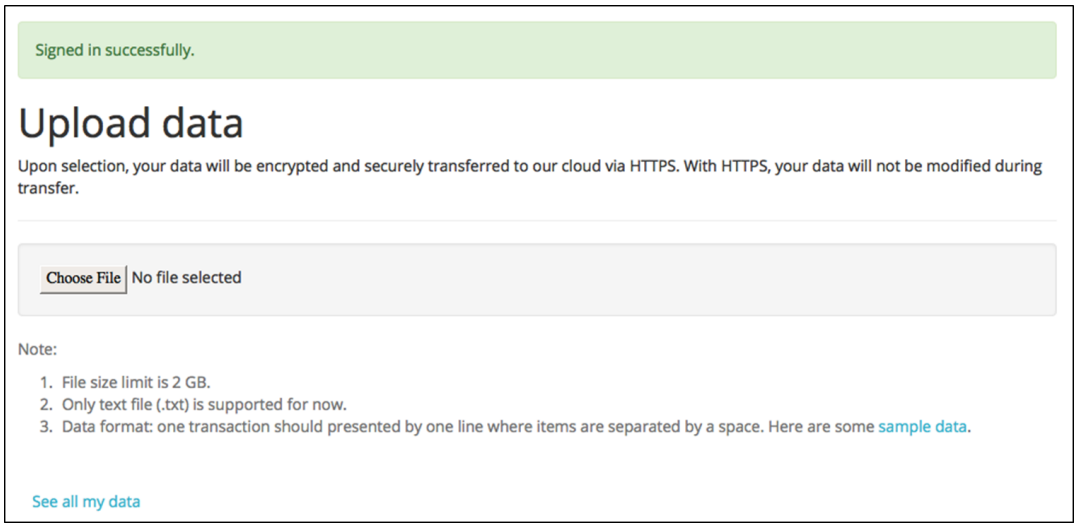


Figure 5: Upload-data page of our FIMaaS system

datasets. To upload this 1.48 GB file in 100 browser tabs at the same time, FIMaaS responded normally. This shows that our FIMaaS system is scalable. The scalability is partially because the Amazon S3 file store offers reliable uploads, and offloads the uploaded task to achieve high availability.

In addition, we also evaluated the transfer speed of this 1.48 GB file from S3 to EC2 on the following two Amazon EC2 instance types:

1. t2.micro[12], which (a) is has the lowest resources and (b) can be served as the baseline of the test; and

2. m1.medium[13], which (a) is the cheapest EC2 instance in EMR platform, (b) is used by FIMaaS worker to mine frequent patterns, (c) can be served as baseline, and is the previous generation of EC2.

Between the two instance types, the transfer speed of t2.micro is observed to be 82 MB/s, while that of m1.medium is only 34.4 MB/s.

4.2 Practicality for Non-expert Miners

To evaluate the effectiveness of FIMaaS web interface, we conducted a user study with 10 participants consisting of 3 data mining experts (who serve as base line) and 7 non-expert miners. The user study mainly includes a set of tasks and a user satisfaction questionnaire.

Recall from Section 3.2 that, after users successfully logged in, they will be redirected to a page to upload their data. So, the participants' first task is to upload the data file—specifically, the online retail file from the Frequent Itemset Mining Dataset Repository—by using the upload-data page as shown in Figure 5. The evaluation results show that all participants correctly uploaded the file. Among them, 8 out of 10 participants (i.e., 73% of the participants) completed this task under 10 seconds with mean time of 7.1 seconds. Between the two groups of participants, non-experts spent about 8.9 seconds whereas experts spent about 3 seconds to complete the task. Those who took longer time were those who read all the text on the upload-data page.

Recall from Section 3.2 that, once the user data are uploaded, users can get the data processed by setting the minimum support *minsup* $\in$ (0.0, 1.0]. So, the participants' second task is to set the input parameter *minsup* for the uploaded online retail file, which captures 88,163 anonymized retail store transactions from an anonymous retail store in Belgium. A sample transaction is {8, 36, 38, 39, 41, 48, 79, 80, 81}, in which each number is an identifier for a merchandise item. Users were instructed to input a *minsup* value indicating that any merchandise item appears in at least 8 transactions are considered to be popular (or frequent). To help users—especially, non-expert miners—to better understand the concepts of minimum support, our FIMaaS system provides a text explanation (as shown in Figure 6) stating that "If you have 100 transactions, then the user-specified minimum support value of 0.1 means that any item appearing in at least $100 \times 0.1 = 10$ transactions is considered to be popular." The evaluation results show that all participants were able to correctly enter $\frac{8}{88,163} \approx 0.00009$ as the input parameter for the *minsup* value. Among them, we observed the following:

- 3 participants (i.e., 30% of the participants) spent less than 1 minute to fill out *minsup* value,

- another 4 participants (i.e., 40%) spent more than 1 minute but less than 2 minutes, and

- the remaining 3 participants (i.e., 30%) spent around 4 minutes to complete this task.

The average completion time for these 10 participants was 2 minutes. Between the two groups of participants, non-experts spent about 2.6 minutes whereas all experts spent less than 1 minutes to complete the task. Again, those who took longer time were those who read all the text on the text explanation on the page.

[12]http://aws.amazon.com/ec2/instance-types/#t2
[13]http://aws.amazon.com/ec2/previous-generation/

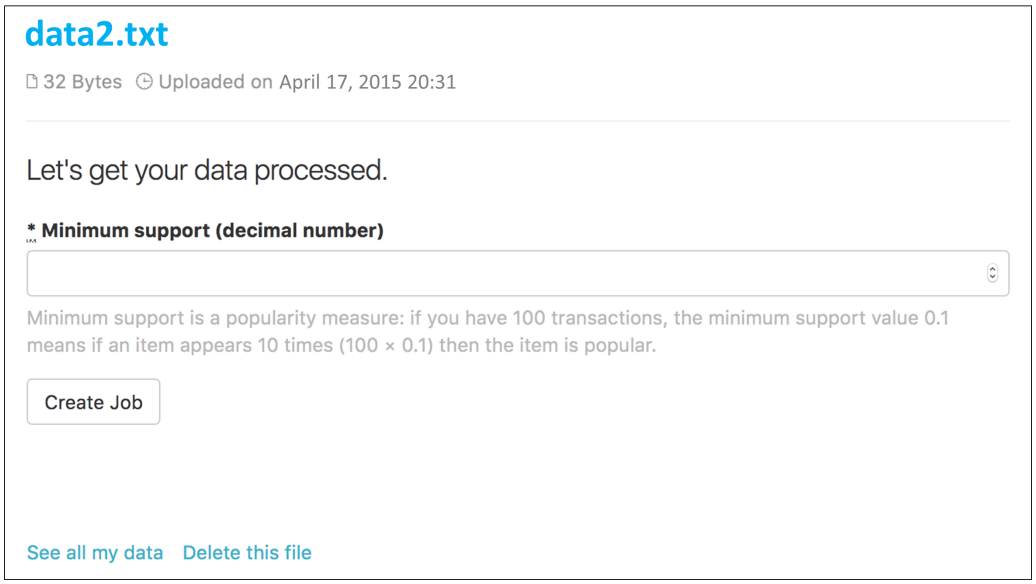


Figure 6: Page for inputting *minsup* to our FIMaaS system

5. CONCLUSIONS

In this paper, we presented the FIMaaS system, which provides users with scalable frequent itemset mining-as-a-service on cloud for non-expert miners. The FIMaaS system also offers a user-friendly user interface, leverages existing cloud solutions to make it scalable to handle large datasets, and uses the same medium as cloud computing, Internet, to make the service accessible from anywhere and any devices.

As an ongoing work, we plan to make FIMaaS more pleasurable for users to use by exploring the following direction. As "a picture is worth a thousand words", we are adding a visualization tool kit, which enables users to visualize frequent patterns. By having a visualization toolkit and integrating it into our proposed FIMaaS system, the users would be able to further exploration and analysis. This leads to the support of *visual analytics*. We are currently exploring the use of either (a) web technologies such as WebGL or (b) the JavaScript visualization frameworks such as D3.js.

6. ACKNOWLEDGEMENTS

This project is partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the University of Manitoba.

7. REFERENCES

[1] R. Agrawal & T. Imielinski, A. Swami. Mining association rules between sets of items in large databases. In Proc. SIGMOD 1993, pp. 207–216.

[2] R. Agrawal & R. Srikant. Fast algorithms for mining association rules. In Proc. VLDB 1994, pp. 487–499.

[3] C. Bettini, X.S. Wang, S. Jajodia & J.-L. Lin. Discovering frequent event patterns with multiple granularities in time sequences. IEEE TKDE, 10(2): 222–237 (1998)

[4] M.-S. Chen, J.S. Park & P.S. Yu. Efficient data mining for path traversal patterns. IEEE TKDE, 10(2): 209–221 (1998)

[5] W.J. Frawley, G. Piatesky-Shapiro & C.J. Matheus. Knowledge discovery in databases: an overview. AI Magazine, 13(3): 57 (1992)

[6] D. Guedes, W. Meira & R. Ferreira. Anteater: a service-oriented architecture for high-performance data mining. IEEE Internet Computing, 10(4): 36–43 (2006)

[7] J. Han, G. Dong & Y. Yin. Efficient mining of partial periodic patterns in time series database. In Proc. ICDE 1999, pp. 106–115.

[8] C.K. Leung. Big data mining applications and services. In Proc. BigDAS 2015, pp. 1–8.

[9] H. Li, Y. Wang, D. Zhang, M. Zhang & E.Y. Chang. PFP: parallel FP-growth for query recommendation. In Proc. ACM RecSys 2008, pp. 107–114.

[10] S. Madden. From databases to big data. IEEE Internet Computing, 16(3): 4–6 (2012)

[11] B. Mobasher, R. Cooley & J. Srivastava. Automatic personalization based on web usage mining. CACM, 43(8): 142–151 (2000)

[12] P. Ramya, S. Mohanavalli & S. Sasirekha. Knowledge extracting system for non-expert miners. Proc. ICCNT 2014, pp. 158–161.

[13] S. Sarawagi & S.H. Nagaralu. Data mining models as services on the internet. ACM SIGKDD Explorations, 2(1): 24–28 (2000)

[14] K. Wang, C. Xu & B. Liu. Clustering transactions using large items. In Proc. ACM CIKM 1999, pp. 483–490.

[15] O.R. Zaïane & J. Luo. Web usage mining for a better web-based learning environment. In Proc. CATE 2001, pp. 60–64.

[16] M. Zorrilla & D. García-Saiz. A service oriented architecture to provide data mining services for non-expert data miners. Decision Support Systems, 55(1): 399–411 (2013)