A Metagenomic Content and Knowledge Management Ecosystem Platform

Binh Vu
FernUniversität in Hagen
Hagen, Germany
binh.vu@fernuni-hagen.de

Yanxin Wu
Cork Institute of Technology
Cork, Ireland
yanxin.wu@mycit.ie

Haithem Afli
Cork Institute of Technology
Cork, Ireland
haithem.afli@cit.ie

Paul Mc Kevitt
Research Institute for Telecommunication and Cooperation
Dortmund, Germany
pmckevitt@ftk.de

Paul Walsh
Nsilico Life Science Ltd.
Cork, Ireland
paul.walsh@nsilico.com

Felix Engel
Research Institute for Telecommunication and Cooperation
Dortmund, Germany
fengel@ftk.de

Michael Fuchs
Wilhelm Büchner University of Applied Sciences
Pfungstadt, Germany
michael.fuchs@wb-fernstudium.de

Matthias Hemmje
FernUniversität in Hagen
Hagen, Germany
matthias.hemmje@fernuni-hagen.de

Abstract—**The reduced cost of DNA sequencing allows metagenomics to be applied on a larger scale. With metagenomic analysis, we have better insight into supplement usage, methane production, and feed conversion efficiency in livestock systems. Nevertheless, sequencing machines generate an enormous amount of complex data. Conventional methods used in the analysis of genomic data involve pre-processing and synchronous reconstruction by multiple systems, which is time consuming and prone to failure. Furthermore, the sequencing datasets and analysis results need to be organized and stored properly in order for scientists to search and access them. To tackle these challenges, a new workflow for metagenomic analysis with improved infrastructure is needed. The MetaPlat project supports experts in both academic and non-academic sectors dealing with challenges in the field of metagenomics by focusing on improved hardware and software platforms. High-performance, fault-tolerant, flexible, and scalable processors and analysis systems will help to increase the effectiveness and efficiency of current metagenomics studies. In this paper, we propose such as an infrastructure applying emerging technologies, such as Kafka, Docker, and Hadoop. Details of the infrastructure solution and some preliminary results are also discussed.**

Keywords—metagenomics, cattle rumen, microbial community, sequencing datasets, classification, taxonomy, cloud architecture, visualization, streaming, Kafka, Docker, Hadoop

I. INTRODUCTION

Global warming is a serious problem that impacts everybody. Experts project that by 2100, earth will be at least 8 degrees Fahrenheit warmer [1]. Higher temperatures create an atmosphere that can collect, retain, and drop more water, changing weather patterns in such a way that wet areas become wetter and dry areas, drier [2]. Extreme weather events are more likely to occur. Ruminant livestock, whilst providing high-quality milk and meat products from otherwise indigestible food components, also produce methane, which contributes significantly to global anthropogenic greenhouse gas emissions [3]. With a better understanding of connections between variations in rumen microbial communities and host genetics, we might find a dietary supplement strategy that helps to reduce methane emissions (CH4) in livestock systems, thus further improving cattle productivity [4] [5].

Whilst genomics is concerned with the genes or entire genome of a specific organism, metagenomics is the field that involves investigation of genomic sequences obtained directly from whole microbial communities present in an environment, such as, e.g., water, soil, human body, and cattle following a culture-independent approach [6]. In-depth analysis of metagenomic sequencing data with support of machine learning and other computer science techniques will provide deeper insights into the complex microbiome ecosystem [7]. The MetaPlat project provides an infrastructure to support the analysis of large metagenomic datasets based on a cloud architecture. The project addresses a key problem, which is the lack of easy-to-use and scalable parallel architectures and approaches to deal with the huge number of generated sequences that are produced in metagenomics [8].

The motivation of this paper is to provide a cloud-based infrastructure to support metagenomic analysis of microbial communities, especially in cattle. Furthermore, analysis results and related scientific content will be packaged and classified using this infrastructure. The paper discusses a solution that is high-performance, fault-tolerant, flexible, and scalable. The chosen architecture and technology should be well-integrated and easy to use. Hence, they not only serve experts in academic but also in non-academic sectors. Section II discusses the challenges of metagenomics. Section III describes technologies applied in the infrastructure. A solution along with its architecture and modelling are introduced in section IV. Section V covers some of the results gathered from the developed system. Finally, section VI concludes and discusses future work.

II. PROBLEM STATEMENT AND APPROACH

A single human genome may consume 100 gigabytes of storage space. By 2025, an estimated 40 exabytes of storage capacity will be required for human genomic data [9]. As sequencing machines can generate thousands of gigabytes per day, there is a challenge for existing infrastructures to manage with this volume. We would need not only high-performance processors and analysis systems but also the intermediate systems that deliver the data from sources to targets and storage systems that are large enough for data persistence.

Considerable research is currently conducted in the area of metagenomic analysis. As a result, a considerable volume of,

978-1-7281-1867-3/19/$31.00 ©2019 IEEE

e.g., scientific publications, sequencing datasets, and statistical figures and tables are produced. Related information is distributed and not necessarily situated side by side. For example, there are several services just for publishing scientific content, such as, e.g., Mendeley, SlideShare, and ResearchGate but without an in-depth search, users cannot find the correct content. Furthermore, it is not easy for authors to manage their scientific content on so many different platforms. Hence, it is necessary to provide a single-point-of-access content and knowledge management system where valuable and related content and knowledge are managed and published.

Scientists must organize knowledge on all living things in the natural world. Otherwise, studying the diversity of millions of living things is overwhelming. Classification also helps scientists identify gaps in their research which provides clues on what to investigate next [10]. In genomics as well as metagenomics, genes need to be classified based on their names and symbols in order to be found quickly. Furthermore, classifying individual genes into groups helps researchers describe how genes are related to each other and to publish their insights in scientific papers. Researchers can use gene groups to predict the function of newly identified genes based on their similarity to known genes [11]. Finally, analysis results related to the genes also need to be organized.

In order to classify genes based on their names and symbols, several genomic taxonomies are needed. The problem is that there is a huge number of gene names and symbols. Also, many genes have more than one name or symbol. Beside sequencing data, analysis results also need to be classified as mentioned above. Hence, a taxonomy management system, that can construct and manage large taxonomies, is important for the classification of metagenomic analysis results.

The next challenge is related to the storage of sequencing datasets, research on these datasets, the analysis of results gathered after processing the datasets, and the classification of publications and presentations produced from these results. We must find a means to combine large volumes of content and knowledge into compound scientific asset packages (packaging) and perform this automatically. Otherwise, there are dissipated fragments of related information. This prevents discovery of important data and the relationships between them. Organizing related content, datasets, and classification information into scientific asset packages is a powerful method of systemizing results produced by metagenomic research.

III. State of the Art

In this section we provide an overview of technologies applied in the infrastructure, such as Hadoop, Kafka, Docker, and the Content and Knowledge Management Ecosystem Portal (KM-EP). Apache Kafka serves as a streaming system to transport genomic data from different sources to analysis systems and the asset manager where metagenomic datasets and analysis results are packaged with other scientific content. Apache Hadoop and Docker provide scalable, fault-tolerant, high performance analysis systems, where computing resources can be better relocated. The MetaPlat KM-EP with its components, such as Taxonomy Manager and Asset Manager, help to organize and manage scientific content, classify metagenomic datasets and analysis results, and combine them into scientific asset packages. The described technologies, which are building blocks of the MetaPlat infrastructure, are introduced in the following sections.

A. *Apache Hadoop*

The storage capacity of hard drives has increased considerably. In 1990, one typical drive could store 1.370 MB of data. 20 years later, one terabyte drives are the norm [12]. Hence, it is no longer efficient, or even possible in some cases, to move large volumes of scientific data required for processing across the network at compute time [13]. Furthermore, with multiple hardware involved, the possibility of failure is high [12].

Hadoop was developed by Doug Cutting, the creator of the widely used text search library Apache Lucene, in 2006. Hadoop is a data storage and processing platform that enables large datasets to be processed locally on the nodes of a cluster using a shared nothing approach, where nodes can independently process a much smaller subset of the entire dataset without needing to communicate with each other [13]. In 2008, Hadoop became a priority project at Apache and the term “Apache Hadoop” is nowadays used for the ecosystem for distributed computing and large-scale data, in which other related projects are also included. The initial version of Hadoop are now standalone products named Hadoop Distributed Filesystem (HDF) and MapReduce. Figure 1 shows the Hadoop ecosystem.

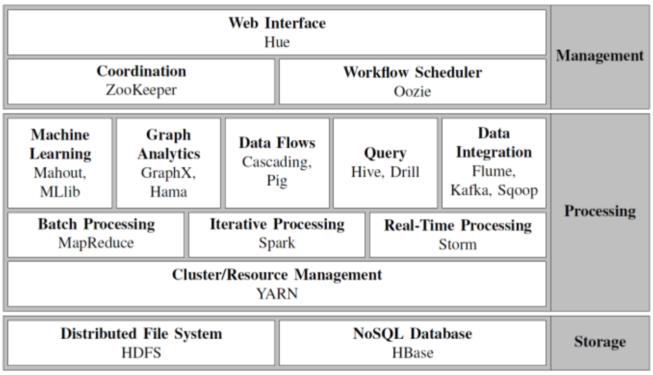


Fig. 1. Constituents of the Hadoop ecosystem shown as a software stack [14]

B. *Apache Kafka*

A technical team from LinkedIn, a social network for professionals, attempted to build a logging system for user activities on their portal using custom in-house components with some support from existing open source tools. The original solution was XML-based logging, to be subsequently processed using different Extract Transform Load (ETL) tools. This approach was not successful and led to various problems. To solve them, LinkedIn developed a system called Kafka [15].

Apache Kafka is a fault-tolerant persistent queuing system, which enables processing large volumes of data in real time [16]. It records messages and organizes them into topics. Applications can then produce or consume messages from these topics [15]. With Kafka, messages are persisted on disk as well as replicated within the cluster to prevent data loss. It can handle hundreds of MBs of reads and writes per second from a large number of clients. The messaging system has cluster-centric design where more nodes can be added to the cluster. Furthermore, many programming languages, such as, e.g., Java, .NET, PHP, Ruby, and Python are supported to facilitate system integration. Finally, messages are moved from producers to consumers immediately, which is a critical requirement for event-based systems [17]. Figure 2 illustrates how Kafka decouples data pipelines.

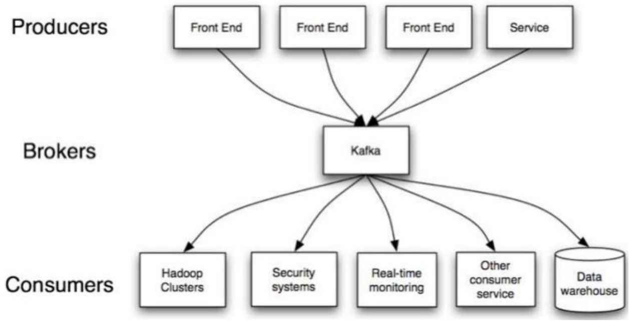


Fig. 2. A scenario supported by Apache Kafka [18]

Producers are the applications that store information in Kafka queues. They send messages to Kafka topics that can store all types of messages. Producers can be a front-end web application that generates logs. They can also perform web analytics that records user activities or a service that sends parameters for an email sender. In contrast, consumers fetch messages from topics and receive information sent by producers. For example, consumers can be a security system that monitors user activities or a database that stores transactions sent by a web application. Hadoop clusters can connect to Kafka as consumers and process big data sent by producers. A typical Kafka cluster consists of multiple brokers, which have the purpose of balancing cluster load. Furthermore, in case of failure, other brokers step in to replace the failed node [15]. This helps Kafka maintain its highly availability and fault-tolerance.

C. Docker

Docker is an open-source project for automating the deployment of applications as portable, self-sufficient containers that can run on the cloud or on-premises. Docker was initiated as an internal project within dotCloud, a platform as a service company [19]. In 2013, Docker was released as open-source to the public [20]. Figure 3 shows the architecture of Docker.

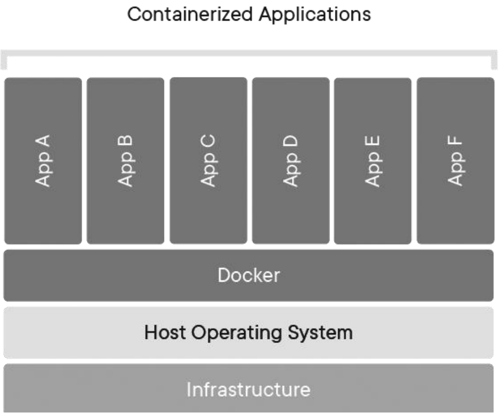


Fig. 3. Applications in separated Docker containers [21]

Docker enables an application to run as expected across different Linux operating systems. The software supports a level of portability that enables a developer to write an application in any language and then readily move it from a laptop to a test or production server regardless of the underlying Linux distribution [19]. To complete this, Docker relies on containers, which are an encapsulation of an application with its dependencies. Unlike virtual machines, containers are more portable and self-contained. Containers not only help application developers but also enable users to download and run applications without needing to expend time on configuration and installation problems or concerns on required system changes [22].

D. Content and Knowledge Management Ecosystem Portal

The Content and Knowledge Management Ecosystem Portal (KM-EP) has been developed to provide a powerful framework for managing knowledge and scientific content. The KM-EP implementation was based on Symfony Framework, which is one of the leading PHP frameworks with a large community, many reusable components, and high-quality documentation. Symfony Framework is used to build applications based on the Model View Controller (MVC) architecture. The controllers are implemented in PHP. MySQL is used to store data of the application in the model layer. jQuery, AngularJS, and Bootstrap help to build and manipulate the user interface in the view layer [23].

The KM-EP consists of five subsystems: (1) Information Retrieval Subsystem (IRS) enabling users to index content and subsequently search for it with support of technologies, such as faceted search. (2) The Learning Management Subsystem (LMS) lets users create online courses without requiring deep knowledge of computer science. Furthermore, LMS helps managing learners, learning materials, and supports the learning process. (3) Content and Knowledge Management Subsystem (CKMS) supports users in the process of creating and managing content and knowledge objects. (4) User Management Subsystem (UMS) manages users and groups along with authentication and authorization of the KM-EP. Finally, (5) Storage Management Subsystem (SMS) enables users to upload and manage files across different cloud services. Figure 4 shows the architecture of the KM-EP.

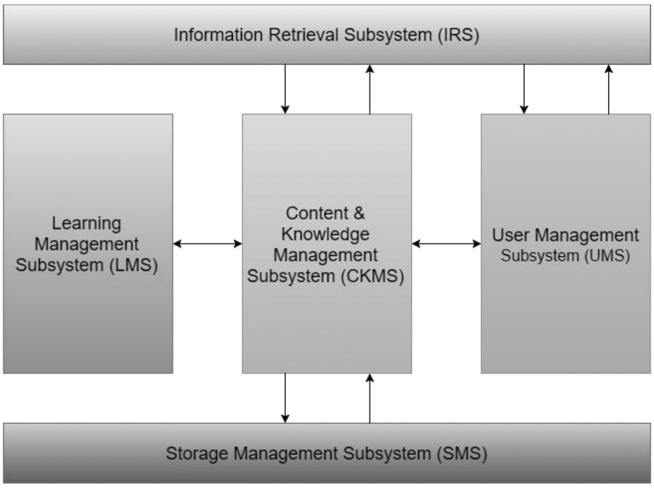

Information Retrieval Subsystem (IRS)
Learning Management Subsystem (LMS)
Content & Knowledge Management Subsystem (CKMS)
User Management Subsystem (UMS)
Storage Management Subsystem (SMS)

Fig. 4. KM-EP architecture [23]

With support of the KM-EP, several R&D projects as well as scientific researches were developed. The platform enables researchers to quickly build prototypes without spending time implementing every component from scratch. Scientific content and knowledge objects can be easily imported to the platform and serve as learning materials for online courses. Popular authentication and authorization methods such as OpenID [24] and OAuth [25] were also included in the KM-

EP. As a result, researchers can integrate their systems with other platforms and exchange information.

Taxonomy Manager is a main component of the KM-EP. Taxonomy Manager is a user-centric system that lets users create and modify their own taxonomies in an easy but effective and efficient way. With the support of crowd sourcing, not only the experts or administrators can build taxonomies, but everyone can join and build their own [23]. Users create their taxonomies from a blank or a seed taxonomy. Other users rate these taxonomies. The highest-rated taxonomy will become the seed for the next round, and it will also be used for content classification and navigation in the system. This process will be repeated until no improvement is required. By using crowdsourcing, users do not have to work in a group and collaborate with other members. The knowledge needed to construct a taxonomy is provided by the wisdom of the crowd. Crowdsourcing is always cheaper than experts. Hence, these taxonomies can be maintained and governed frequently by the crowd [26].

Asset Manager is another important component of the KM-EP. Asset Manager supports users in the creation and management of scientific asset packages. In the Asset Manager, users can manage their own scientific assets. They have options to create new genomic packages, edit existing packages, or remove them from the management system. Whilst creating a new asset, users have the option to insert information related to the asset such as e.g. title, description, authors' name, and organization. Furthermore, users can search for scientific content from the Digital Library to include in this package by typing their titles. Asset Manager searches for content, the title of which contains the typed text, and illustrates the result as a list in real-time. Users can then choose the right content from the result.

IV. Solution Design and Modelling

To overcome challenges in the field of metagenomics, we propose a cloud-based Metagenomic Content and Knowledge Management infrastructure, that is easy to use, scalable, fault-tolerant, and has high performance. Figure 5 shows the architecture of the infrastructure. The infrastructure solution is combined of five different systems that have separate functionalities and apply different technologies. They are described in the following sections.

A. Genomic Data Streaming Mediator

The conventional IT infrastructure to process genomic sequences involves a set of computing nodes accessed through a batch queuing system and equipped with a parallel shared storage system. Whilst it is a working solution, this approach is lacking robustness and scalability [27]. By introducing a high-performance distributed stream-processing system Kafka, genomic sequencing integration and analysis processes can be significantly improved in terms of performance, fault-tolerance, and scalability.

Given the fact that the size of genomic samples is usually very large and the machines can only read them in fragments, a high-performance system is needed to receive results from distributed genomic sequencing machines, pre-process, and reconstruct them into a completed and integrated genomic sequence sample. Furthermore, the input data not always come from a single source but there can be multiple different data sources used in the sequencing process.

In the Genomic Data Streaming Mediator (GDSM) of MetaPlat, Kafka is applied to solve these challenges. Its API enables different data sources to publish their output to the cluster. With each data source, there is a source connector that forwards the data from the source to Kafka's topics. If there is a new source, it can simply register to a topic and start sending records to it. The consumers or the systems that receive these records do not need to know or manage the sources. Hence, it is easier to add or remove clients at both ends. Whilst the sequencing data is sent by data sources (or producers), systems that pre-process the data can receive it with Streams API. This enables clients to obtain records from one or more input topics, perform transformations on these records and publish the transformation results to one or more output topics [14]. The pre-processing system in this case would work with the sequencing data and send them back to Kafka in real-time. Another system, which has the purpose of reconstructing sequencing fragments into a completed sample, would listen to the pre-processing system's topic. As soon as there are results from the pre-processor, it can start re-construct the sequences without waiting for every fragment to arrive.

Furthermore, Kafka also enables real-time and post-analysis systems to subscribe to its topics and receive sequencing data for further analysis using the Connect API. Analysis results can be sent back to Kafka for other applications to display or storage.

B. Metagenomic Analysis Cluster

The Metagenomic Analysis Cluster (MAC) of MetaPlat provides a means of scaling up the computing infrastructure. Hence, it leads to higher availability as well as fault-tolerance and load balancing by using Docker swarm and Hadoop clusters.

With Docker, analysis applications can be deployed automatically in separate containers. It is easier to replicate the application into multiple instances and lets them run parallel. For MetaPlat, sequencing data is sent by data sources to the Kafka cluster. Subsequently, they are obtained by duplicated instances of the analysis application, which take separate fragments and process them independently. These containers form a Swarm cluster and are managed by a Swarm manager. The manager can fill the least utilized machines with containers or ensures that each machine gets exactly one instance of the specified container [28]. Hence, Swarm clusters are especially useful in case of resource utilization or when one or more nodes experience an outage.

Hadoop clusters are a special type of computational cluster. They are used in the MetaPlat for storing and analyzing large amounts of genomic data, which can be structured or unstructured. Asset Manager service would submit analysis jobs to the Hadoop cluster through Kafka's topics. Data is also pulled from Kafka to the cluster. The JobTracker inside the Hadoop cluster then farms out tasks to nodes in the cluster and pushes results back to Kafka topics after the work is completed.

When the workload increases, more Kafka brokers can easily be added, and nodes to the Docker swarm and Hadoop cluster. Docker swarm and Hadoop cluster are usually set up to work independently, where both of them are installed on multiple bare metal servers. Nevertheless, there are problems with managing Hadoop on fixed physical infrastructure, such as under-utilization, duplication of data, time consumption,

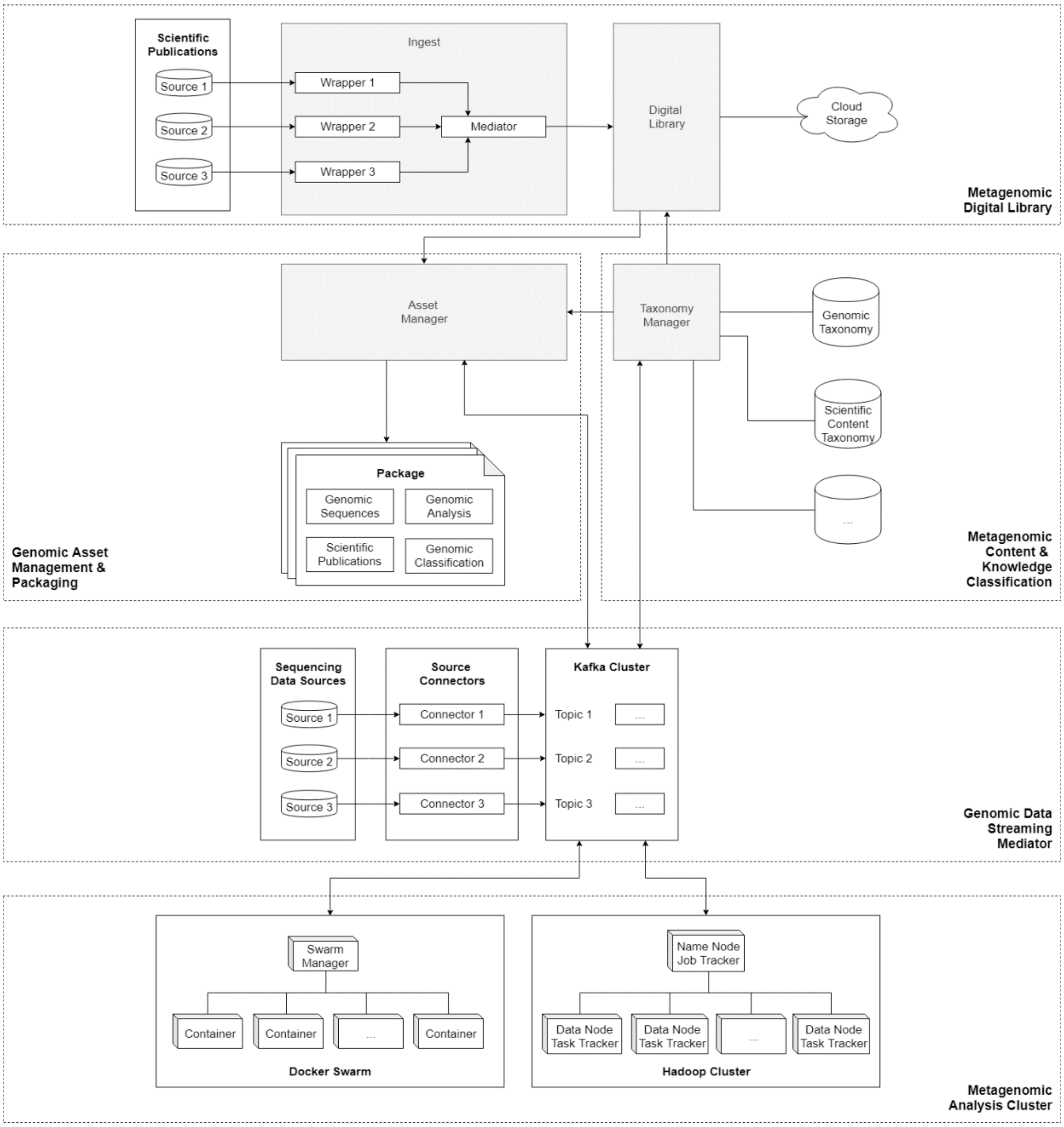


Fig. 5. Architecture of the Metagenomic Content and Knowledge Management infrastructure

and management complexity. Here, Hadoop nodes were deployed in Docker containers. Through this, we have the flexibility of virtualization whilst maintaining power of bare metal performance. New Hadoop nodes can be quickly deployed and configurated. Furthermore, they can be added on the least utilized machines, which reduces resource wastage. Finally, created resources can be dynamically relocated without downtime.

C. *Metagenomic Digital Library*

The KM-EP component Ingest enables scientific content to be imported into the MetaPlat KM-EP. Using a Mediator-Wrapper Architecture, content from various data sources, such as Mendeley, SlideShare, and in different formats, such as BibTex [29] and OAI-PMH [30], can be queried, uploaded, and integrated into the Metagenomic Digital Library (MDL) of MetaPlat. The wrapper layer overcomes the heterogeneity problem by fetching and transforming the data from the sources to a universal format. Mediator acts as a middleware sending requests to wrappers and merging results obtained from them.

The Ingest provides a plugin for each data source. This plugin includes a user interface and necessary functionalities that enable users to interact with the data source, such as browsing through the content list, searching for content, and selecting the candidate they wish to import. In the case of BibTex, the plugin lets users upload the file and select entries from the list extracted from the file. Furthermore, in some services, users need to log-in to their account in order to select content. In this case, the corresponding plugin has to support the authentication and authorization features of the source. For example, Mendeley requires client applications to connect to their server using the authentication framework OAuth2. After user login, the system returns an access token to the application, so it can use this token to provide authorization for API requests.

After content is selected and the import process is initiated, wrappers download selected content from their data sources. Presently, this content is at this moment are in different data formats depend on its source. It is the wrapper's job to convert it into a chosen format that can be understood by the mediator.

Scientific content metadata, such as publications or presentations, needs many fields to describe their properties. Meanwhile, entries in OAI-PMH or BibTex have a limited number of fields, which sometimes cannot be matched to scientific content fields. Furthermore, each data source has a large number of records, which leads to the requirement of importing thousands of records from users. If the system coverts data to the universal format automatically and it maps the fields incorrectly, users may have to fix errors by hand in thousands of records. On the other hand, if the Ingest cannot do it automatically and users have to map fields for each record, this would also consume much effort. To solve this problem, we introduced a semi-automatic import process, where the wrappers automatically map fields that they know and leave the unknown fields for users to handle. Each time there is a field that is new to the system, it will ask users to decide how to map it at the beginning of the import process. By doing so, users do not have to configure the wrappers every time they need to import and are still able to change the mapping if it is incorrect.

D. *Metagenomic Content and Knowledge Classification*

In the Metagenomic Content and Knowledge Classification (MCKC) of MetaPlat, the KM-EP's component Taxonomy Manager supports the development of genomic taxonomies. This type of taxonomy normally contains a large number of terms. Working on a large tree of thousands of nodes needs a lot of resources, such as computing time, memory, and disk space. Multiply that to hundreds to thousands of users who use the system at the same time and there will be considerable resources required. With fast and efficient algorithms, the KM-EP's Taxonomy Manager helps to speed up the management process whilst requiring fewer resources. Together with caching mechanisms, the Taxonomy Manager is required for users to work on large genomic taxonomies.

The module Categorization of the Taxonomy Manager is used in the KM-EP to classify scientific content and knowledge to different categories. By breaking a large set of documents into smaller subsets, classification makes them easier to manage and sort. Meanwhile, documents often have thousands of words. This makes searching with keywords not always possible as a word may have several meanings in a given language. Hence, classifying documents by topics helps finding them faster. In the case of multiple matches, the result can be filtered out based on the classification topics. Furthermore, other documents on the similar topics can be suggested to users as relevant content. In MetaPlat, component Categorization supports classification of scientific content in the MDL and genomic packages managed by the Asset Manager using genomic taxonomies and scientific content taxonomies developed by the Taxonomy Manager. Furthermore, metagenomic analysis results produced by Docker swarm and Hadoop cluster are also classified using genomic taxonomies in order to support easy access and improved research scholarship.

When users access content and knowledge through the MCKC in the KM-EP, a categorization panel is shown in the UI. All main taxonomies are listed on the panel. After users select a taxonomy, the taxonomy with its terms is loaded and presented in tree format to them. Users can then decide which terms are associated with the current content and assign them to the content by selecting terms on the tree. The selection is submitted to the back-end. The back-end receives the request and attempts to insert the categorization entry into the database. The back-end also notifies the search server for change in the categorization of the content. Hence, indexed data can be updated and new categorizations are presented immediately in search results.

Each term and the MCKC taxonomy itself have a persistent identifier (PID), which helps to distinguish them from each other. The system uses this PID to detect cloned versions of a base taxonomy and to maintain the persistence of the classification during evolution of a taxonomy. Even in the case where a term's content was changed completely, the classifications using this term will not be lost. Furthermore, category information in the taxonomy, such as their PID and path, are indexed by the search server along with the MCKC and categorization of content. By collecting this information, the taxonomy can be used to filter search results. With the faceted search, users have an overview of the classification of content in real-time and quickly find search results by selecting only relevant topics. In the case of system navigation, after pages are classified by taxonomy categories, users can navigate to the web-page by filtering categories, which is similar to what they have done whilst searching for content.

E. *Genomic Asset Management and Packaging*

The Genomic Asset Management of Packaging (GAMP) component is where related content, datasets, and analysis results are gathered within the MetaPlat KM-EP and combined into packages. Genomic sequences can be added to the package by users. This process is similar to adding scientific content, where users enter the title of the content and choose the correct one from the result list. The genomic sequences can also be added to the package automatically by other services running in the background. One example is the service that receives data from the Kafka cluster. This service works as a daemon that runs in the background. It has a sink connector built in, so it can subscribe to data topics on the Kafka cluster. These topics are polled at particular times, usually every 30 minutes, for new messages which contains genomic sequences. Based on their metadata, genomic sequences are automatically added to the corresponding GAMP packages. By using this service, new data can be added to the GAMP package automatically without user effort. This is a much more realistic approach since modern sequencing machines produce terabytes of data per day.

Genomic sequences are used for real-time and post analysis. With real-time analysis, the sequences are extracted from the Kafka cluster by Docker swarms and Hadoop clusters. They are processed and transformed by containers and data nodes inside these distributed systems, which guarantees low latency, fault-tolerance, scalability, and flexibility. Finally, the results are returned to Kafka. In this case, another service of the GAMP will pull the metagenomic analysis results from Kafka and add them to their packages. In the case of post-analysis, there is no need to process the sequences in real-time. Hence, processing services can run as background daemon and store sequences in the packages. Performing analysis in batch-mode, the services require less resources than in real-time processing, which requires a High-Performance-Computing infrastructure. Nevertheless, the sequences can also be extracted from the packages and sent to big data post-analysis systems using the Kafka cluster.

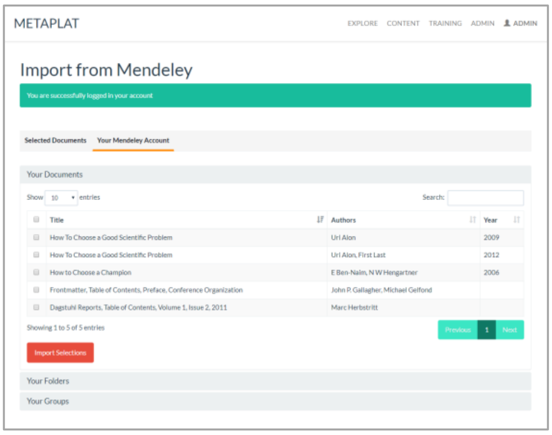
With support of the Taxonomy Manager, genomic sequences, scientific content, and metagenomic analysis results inside a GAMP package can be classified using the developed genomic, scientific content, and other taxonomies. Furthermore, not only the content inside a package but also the GAMP package itself can be classified into different categories using the Taxonomy Manager. This can be achieved by the integration of the Categorization module into GAMP. Users have the option to classify the package and content inside it by selecting the relevant categories in the user interface. The classification information is stored in the genomic package as well as indexed in the search server.

With GAMP, relevant content, genomic datasets, analysis results, and genomic classifications can be gathered and managed in one central location. Hence, it takes less effort to deploy and maintain them. Furthermore, due to the centralization of content and knowledge, as well as classification support, relevant information can be found quickly and easily by users.

V. PRELIMINARY RESULTS

Based on the solution models and designs introduced in the previous sections, the MetaPlat infrastructure has been developed with preliminary results. The infrastructure can be described in three main systems: (1) the MetaPlat KM-EP, which is comprises the MDL, GAMP, and MCKC; (2) the GDSM, and (3) the MAC.

The MetaPlat KM-EP supports the scientific content ingesting and management process. The KM-EP Taxonomy Manager component helps to develop genomic and content classification taxonomies. The Asset Manager component gathers and includes genomic sequences and genomic datasets, scientific content, and metagenomic analysis results into packages. Furthermore, these packages and their content are classified with support of the Taxonomy Manager. Figure 6 shows the MetaPlat KM-EP.



Import from Mendeley

You are successfully logged in your account

Your Documents

Show 10 entries

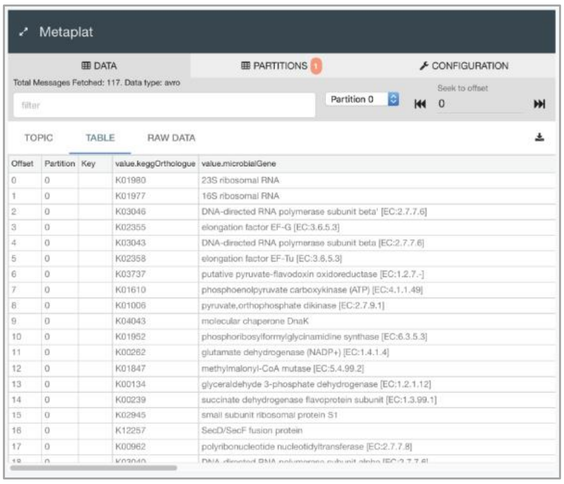
Search:

□	Title	Authors	Year
□	How To Choose a Good Scientific Problem	Uri Alon	2009
□	How To Choose a Good Scientific Problem	Uri Alon, First Last	2012
□	How to Choose a Champion	E Ben-Naim, N W Hengartner	2006
□	Frontmatter, Table of Contents, Preface, Conference Organization	John P. Gallagher, Michael Gelfond	
□	Dagstuhl Reports, Table of Contents, Volume 1, Issue 2, 2011	Marc Herbstritt	

Showing 1 to 5 of 5 entries

Fig. 6. MetaPlat KM-EP user interface

The Kafka cluster in the GDSM enables sequencing data to be streamed from the data sources, pre-processed, and reconstructed by processing services and applications. Furthermore, Kafka helps to move genomic data between the GAMP Asset Manager and analysis systems in the ecosystem. On one hand, the GAMP Asset Manager can download the constructed sequences and store them in their packages. On the other hand, analysis systems can receive the stored sequences for these GAMP packages for further post-analysis. The results are also sent back to the GAMP Asset Manager using the same Kafka cluster. Figure 7 presents the sequencing data published to different topics on the Kafka cluster.

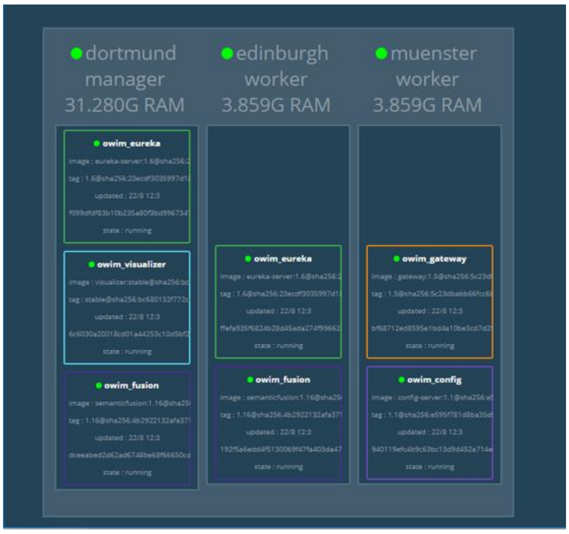


Total Messages Fetched: 117. Data type: avro

Offset	Partition	Key	value.keggOrthologue	value.microbialGene
0	0		K01980	23S ribosomal RNA
1	0		K01977	16S ribosomal RNA
2	0		K03046	DNA-directed RNA polymerase subunit beta' [EC:2.7.7.6]
3	0		K02355	elongation factor EF-G [EC:3.6.5.3]
4	0		K03043	DNA-directed RNA polymerase subunit beta [EC:2.7.7.6]
5	0		K02358	elongation factor EF-Tu [EC:3.6.5.3]
6	0		K03737	putative pyruvate-flavodoxin oxidoreductase [EC:1.2.7.-]
7	0		K01610	phosphoenolpyruvate carboxykinase (ATP) [EC:4.1.1.49]
8	0		K01006	pyruvate,orthophosphate dikinase [EC:2.7.9.1]
9	0		K04043	molecular chaperone DnaK
10	0		K01952	phosphoribosylformylglycinamidine synthase [EC:6.3.5.3]
11	0		K00262	glutamate dehydrogenase (NADP+) [EC:1.4.1.4]
12	0		K01847	methylmalonyl-CoA mutase [EC:5.4.99.2]
13	0		K00134	glyceraldehyde 3-phosphate dehydrogenase [EC:1.2.1.12]
14	0		K00239	succinate dehydrogenase flavoprotein subunit [EC:1.3.99.1]
15	0		K02945	small subunit ribosomal protein S1
16	0		K12257	SecD/SecF fusion protein
17	0		K00962	polyribonucleotide nucleotidyltransferase [EC:2.7.7.8]
18	0		K03040	DNA-directed RNA polymerase subunit alpha [EC:2.7.7.6]

Fig. 7. Kafka topics' user interface [31]

The third system is the MAC based on Docker swarm and Hadoop cluster. These systems help to provide powerful analysis as well as to process very large volumes of structure and unstructured data coming from the Kafka cluster. The computing cluster can be scaled up in case of increasing workload. Furthermore, in case of an outage, the swarm manager will create new containers to replace the failed one. Figure 8 shows the analysis system running on a Docker swarm, which consists of a swarm manager and two workers.



owim_eureka
image : eureka-server:1.6@sha256:2
tag : 1.6@sha256:23ecdf3035997d1
updated : 22/8 12:3
f099dfdf83b10b235a80f3bd996734
state : running

owim_visualizer
image : visualizer:stable@sha256:bc
tag : stable@sha256:bc680132f772c
updated : 22/8 12:3
6c6030a20018cd01a44253c10d5bf2
state : running

owim_fusion
image : semanticfusion:1.16@sha25
tag : 1.16@sha256:4b2922132afa371
updated : 22/8 12:3
dceeabed2d62ad6748be68f66650cd
state : running

owim_eureka
image : eureka-server:1.6@sha256:2
tag : 1.6@sha256:23ecdf3035997d1
updated : 22/8 12:3
ffefa935f6824b28d45ada274f99662
state : running

owim_fusion
image : semanticfusion:1.16@sha25
tag : 1.16@sha256:4b2922132afa371
updated : 22/8 12:3
192f5a6edd4f5130069f47fa403da47
state : running

owim_gateway
image : gateway:1.5@sha256:5c23d
tag : 1.5@sha256:5c23dbabb66fcc6
updated : 22/8 12:3
bf68712ed8595e1bd4a10be3cd7d2
state : running

owim_config
image : config-server:1.1@sha256:e
tag : 1.1@sha256:e595f781d8ba35d
updated : 22/8 12:3
940119efc4b9c63bc13d9d452a714e
state : running

Fig. 8. Analysis systems running on a Docker swarm [32]

VI. SUMMARY AND FUTURE WORK

Here, metagenomics has been introduced as an approach to fight global warming by analyzing the microbial communities in cattle. As a result, a dietary supplement strategy for ruminant livestock can be discovered to help reduce the methane emissions. The analysis of metagenomic sequencing data is being supported by machine learning and other computer science techniques. Nevertheless, this poses several challenges such as datasets transfer and storage, scientific content management, and information classification. As an approach to solve these challenges, we propose a cloud-based Metagenomic KM-EP infrastructure to support the analysis of large metagenomic datasets within the MetaPlat project. The solution is a combination of different systems, such as the Kafka cluster for genomic streaming pipeline, Docker swarm and Hadoop cluster for genomic processing and analysis. Furthermore, the content and knowledge management ecosystem KM-EP has an important role in scientific content management, genomic taxonomy evolution and management, and genomic classification and asset packaging. All these systems provide experts in both academic and non-academic sectors a cloud-based infrastructure that is easy to use, fault-tolerant, flexible, scalable, and has high-performance. A prototype has been developed which gives some preliminary promising results. They have shown that whilst challenging, developing such an infrastructure is possible. The MetaPlat KM-EP works along with the Kafka cluster and Docker Swarm to receive sequencing datasets, and processes, classifies, and gathers them into genomic packages. Future work will include exploding the potential of Docker and Hadoop in developing asynchronous processors and analysis systems for metagenomic.

ACKNOWLEDGMENT

This publication has been produced in the context of the MetaPlat project. This project has received funding from the European Union's H2020 Programme under grant agreement No. 690998. However, this paper reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

REFERENCES

[1] J. Walsh and D. Wuebbles, "Our Changing Climate," National Climate Assessment, 2014.

[2] M. Denchak, "Are the Effects of Global Warming Really that Bad?," 15 March 2016. [Online]. Available: https://www.nrdc.org/stories/are-effects-global-warming-really-bad. [Accessed 9 September 2019].

[3] H. Wang, H. Zheng, R. J. Dewhurst and R. Roehe, "Microbial Co-presence and Mutual-exclusion Networks in the Bovine Rumen Microbiome," in *2017 IEEE International Conference on Bioinformatics and Biomedicine*, 2017.

[4] H. Wang, H. Zheng, F. Browne, R. Roehe, R. J. Dewhurst, F. Engel, M. Hemmje, X. Lu and P. Walsh, "Integrated Metagenomic Analyses of the Rumen Microbiome of Cattles Reveals Key Biological Mechanisms Associated with Methane Traits," *Methods*, pp. 108-119, 2017.

[5] J. T. Wassan, H. Zheng, F. Browne, J. Bowen, P. Walsh, R. Roehe, R. Dewhurst, C. Palu, B. Kelly and H. Wang, "An Integrative Framework for Functional Analysis of Cattle Rumen Microbiomes," in *2018 IEEE International Conference on Bioinformatics and Biomedicine*, 2018.

[6] T. Manning, J. T. Wassan, C. Palu and H. Wang, "Phylogeny-Aware Deep 1-Dimensional Convolutional Neural Network for the Classification of Metagenomes," in *2018 IEEE International Conference on Bioinformatics and Biomedicine*, 2018.

[7] J. T. Wassan, H. Wang, F. Browne and H. Zheng, "A Comprehensive Study on Predicting Functional Role of Metagenomes Using Machine Learning Methods," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pp. 751-763, 2019.

[8] P. Walsh, C. Palu, B. Kelly, B. Lawor, J. T. Wassan, H. Zheng and H. Wang, "A Metagenomics Analysis of Rumen Microbiome," in *2017 IEEE International Conference on Bioinformatics and Biomedicine*, 2017.

[9] The Medical Futurist, "The Genomic Data Challenges Of The Future," 27 October 2018. [Online]. Available: https://medicalfuturist.com/the-genomic-data-challenges-of-the-future.

[10] M. Welch, Why is it Important to Classify Living Things?, 2002.

[11] "What are gene groups?," 3 September 2019. [Online]. Available: https://ghr.nlm.nih.gov/primer/genefamily/genefamilies. [Accessed 9 September 2019].

[12] T. White, Hadoop: The Definitive Guide, O'Reilly Media, Inc., 2012.

[13] J. Aven, Hadoop in 24 Hours, Sams Teach Yourself, Sams Publishing, 2017.

[14] K.-D. Schmatz, K. Berwind, F. Engel and M. L. Hemmje, "An Interface to Heterogeneous Data Sources Based on the Mediator/Wrapper Architecture in the Hadoop Ecosystem," in *2018 IEEE International Conference on Bioinformatics and Biomedicine*, 2018.

[15] M. Kumar and C. Singh, Building Data Streaming Applications with Apache Kafka: Designing and deploying enterprise messaging queues, Packt Publishing, 2017.

[16] S. Minni, Apache Kafka Cookbook, Packt Publishing Ltd, 2015.

[17] N. Garg, Learning Apache Kafka, Packt Publishing Ltd, 2015.

[18] J. Stein, "Current and Future of Apache Kafka," 13 November 2014. [Online]. Available: https://www.slideshare.net/charmalloc/current-and-future-of-apache-kafka.

[19] N. Martin, "A brief history of Docker Containers' overnight success," 11 May 2015. [Online]. Available: https://searchservervirtualization.techtarget.com/feature/A-brief-history-of-Docker-Containers-overnight-success. [Accessed 9 September 2019].

[20] C. Tozzi, "Docker at 4: Milestones in Docker History," 23 March 2017. [Online]. Available: https://containerjournal.com/features/docker-4-milestones-docker-history/. [Accessed 9 September 2019].

[21] Docker, "What is a Container?," [Online]. Available: https://www.docker.com/resources/what-container. [Accessed 9 September 2019].

[22] A. Mouat, Using Docker, O'Reilly Media, Inc., 2015.

[23] B. Vu, J. Mertens, K. Gaisbachgrabner, M. Fuchs and M. Hemmje, "Supporting Taxonomy Management and Evolution in a Web-based Knowledge Management System," in *HCI 2018*, Belfast, UK, 2018.

[24] "Specifications," OpenID, [Online]. Available: https://openid.net/developers/specs/. [Accessed 28 October 2019].

[25] "OAuth 2.0," Oauth Community Site, [Online]. Available: https://oauth.net/2/. [Accessed 28 October 2019].

[26] B. Vu and M. Hemmje, "Supporting Taxonomy Development and Evolution by Means of Crowdsourcing," in *International Conference on Knowledge Engineering and Ontology Development*, Wien, 2019.

[27] F. Versaci, L. Pireddu and G. Zanetti, "Kafka Interfaces for Composable Streaming Genomics Pipelines," in *2018 IEEE EMBS International Conference on Biomedical & Health Informatics*, 259-262, 2018.

[28] Docker, "Get Started, Part 4: Swarms," [Online]. Available: https://docs.docker.com/get-started/part4/. [Accessed 9 September 2019].

[29] "Your BibTeX resource," BibTeX, 2016. [Online]. Available: http://www.bibtex.org/. [Accessed 28 October 2019].

[30] "Protocol for Metadata Harvesting," Open Archives Initiative, [Online]. Available: https://www.openarchives.org/pmh/. [Accessed 28 October 2019].

[31] K.-D. Schmatz, "Konzeption, Implementierung und Evaluierung einer datenbasierten Schnittstelle für heterogene Quellsysteme basierend auf der Mediator-Wrapper-Architektur innerhalb eines Hadoop-Ökosystems," FernUniversität in Hagen, Hagen, 2018.

[32] R. Donovan, M. Healy, H. Zheng, F. Engel, B. Vu, M. Fuchs, P. Walsh, M. Hemmje and P. M. Kevitt, "SenseCare: Using Automatic Emotional Analysis to Provide Effective Tools for Supporting," in *IEEE International Conference on Bioinformatics and Biomedicine*, 2018.