Investigating the Performance of Moodle Database Queries in Cloud Environments

Karina Wiechork[1,2][a] and Andrea Schwertner Charão[2][b]
[1]*Information Technology Coordination, Federal Institute of Education Science and Technology Farroupilha, Frederico Westphalen, Brazil*
[2]*Department of Languages and Computer Systems, Federal University of Santa Maria, Santa Maria, Brazil*

Keywords: Cloud Computing, Benchmarks, Database, Performance.

Abstract: Several computing services are being migrated to cloud environments, where resources are available on demand and billing is based on usage. Databases in the cloud are increasingly popular, however their performance is a key indicator that must be known before deciding to migrate a system to a cloud environment. In this article, we present a preliminary investigation of the performance of database queries in Moodle, a popular Learning Management System, installed on cloud environments from Amazon Web Services and Google Cloud Platform. Experiments and performance analysis were based on benchmarks Pgbench, Sysbench and a Moodle Benchmark Plugin. We collected data and compare it with the results obtained on a local computer. In the configurations we tested, the results show that the Moodle database at Amazon's provider performed better than Google's. We made our data and scripts available to favour reproductibility, so to support decision makers on the migration of a Moodle instance to a cloud service provider.

1 INTRODUCTION

Cloud computing is not only changing the way we use software, but also the way we build it. As more and more services migrate to the cloud, traditional components in software architecture may be provided by cloud-based services. Since the early days of cloud computing, the range of services has grown and spanned from personal to corporate applications. Examples of cloud services include word processors, spreadsheets, database managers and a lot more (Miller, 2008).

The use of cloud computing platforms is advantageous for a number of factors, among then: scalability, which is a prime factor for distributed and high-performance applications, the ease of configuring the instances for running applications, and the elimination of the initial cost to acquire and operate the required infrastructure.

Data management is a very important factor within the context of organizations that keep all their data in computerized environments. Data security, scalability, and performance are all aspects that need to be guaranteed in cloud computing environments.

There are many providers offering cloud-based data management, including traditional, relational database management systems (RDBMS). Many organizations rely on software built around traditional RDBMS, as for example educational institutions and their Learning Management Systems. Migrating such systems to a cloud environment may be advantageous, but is not a trivial decision.

In this article, we present a preliminary investigation of the performance of database queries over Moodle (Modular Object-Oriented Dynamic Learning Environment), a popular Learning Management System (LMS), in cloud environments from distinct providers. This open-source LMS is actively supported and is widely used by many educational institutions all over the world. Despite its popularity, there is a lack of studies focusing on its performance on cloud environments.

We decided to focus on the Moodle database because it is a major component within this LMS and it depends on a third-party RDBMS (PostgreSQL). We chose two major cloud storage providers, Amazon Web Services (AWS) and Google Cloud Platform (GCP), which offer a wide range of pricing plans, including free cloud tiers. Performance was assessed

[a] https://orcid.org/0000-0003-2427-1385
[b] https://orcid.org/0000-0003-3695-8547

Wiechork, K. and Charão, A.
Investigating the Performance of Moodle Database Queries in Cloud Environments.
DOI: 10.5220/0009792202690275
In *Proceedings of the 22nd International Conference on Enterprise Information Systems (ICEIS 2020) - Volume 1*, pages 269-275
ISBN: 978-989-758-423-7

using different benchmark tools. The results and our reproductible process can be useful to support decision makers on the migration of a Moodle instance to a cloud service provider. The obtained results were compared with compared with the experiments results using the local installation of Moodle.

The rest of this paper is organized as follows. Section 2 discusses some background topics and the related works that comprised experiments involving database performance. In Section 3, we present the methodology and the environments we used. Section 4 details the experiments and the results we obtained. Lastly, section 5 concludes this paper with a brief summary and suggestions for future research.

2 BACKGROUND AND RELATED WORK

2.1 Moodle

Moodle is an e-learning platform based on free software, it is known as LMS, as Virtual Learning Environment (AVA) or Course Management System (CMS). This platform offers mechanisms for sharing digital content, as well as communication tools, creating courses, monitoring user actions that take place during a course, and can be enriched with different plugins, designed to meet the specific needs of a given set of users. It is also possible, through this tool, to develop evaluative activities, such as task delivery, questionnaires, among other actions in distance learning activities.

Moodle is a social construction education framework and can be run on any computer that has a Database Management System compatible with SQL (Structured Query Language). The Moodle 3.8 database includes about 422 tables. Also, it features a database abstraction layer called XMLDB, that is, Moodle's working code is the same in Maria DB, MS SQL Server, MySQL, Oracle and PostgreSQL. Based on the documentation on the official Moodle website, PostgreSQL is the preferred engine to host the Moodle database with large tables (Moodle, 2016).

In addition to our work, other research has also investigated the performance of the Moodle environment, but in other areas. In the work Caminero et.al (Caminero et al., 2013), tests the performance of three LMS programs: LRN (Learn, Research, Network), Sakai and Moodle. This study was concerned with collecting measurements of memory and CPU consumption. In the research Guo et.al (Guo et al., 2014), the performances were compared with Moodle in a physical and virtualized environment in cloud computing, where the authors used the Siege pressure test tool.

2.2 Cloud Computing Models

Cloud computing is a convenient abstraction of virtualized computing resources, including hardware and software, that are accessible over a network. Cloud-based solutions have changed the way individuals and organizations deal with the computing resources to meet their needs. Instead of investing on hardware and software maintenance, they may rely on pay-as-you-go, scalable, cloud-based alternatives (Jamsa, 2013). Given the wide range of cloud computing options, different models and taxonomies have been proposed. In this work, we have initially considered three widely accepted cloud service models proposed by the National Institute of Standards and Technology (NIST): Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) (Mell and Grance, 2011).

Infrastructure as a Service (IaaS). Delivers a set of virtual machines with associated storage, processors, network connectivity and other relevant resources. Rather than purchasing all required equipment, consumers lease such resources as part of a fully outsourced service (Mahmood and Hill, 2011). Examples of this service are: Amazon Web Services, Google Cloud Platform, Microsoft Azure, among others.

Platform as a Service (PaaS). The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure (Mell and Grance, 2011). PaaS also allows you to avoid spending and the complexity of buying and managing software licenses or development tools and other resources.

Software as a Service (SaaS). The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities (Mell and Grance, 2011). This is the case, for example, of the package Microsoft Office 365, Onedrive, Dropbox, Google Drive, e-mail services, among others.

Both IaaS and SaaS models are well suited for migrating Moodle to the cloud. Indeed, there are some companies offering a Moodle environment in a SaaS model, preconfigured and ready to use, such as: AWS Marketplace (Amazon, 2013), Bitnami (Bitnami, 2020), MoodleCloud (Moodle, 2019), among others. On the other end, the IaaS model offers more autonomy over the server, so this is the approach we adopted in this work. We have chosen the providers from Amazon and Google, two prominent companies which offer a wide range of pricing plans, including free instances which are convenient for preliminary experiments.

2.3 Cloud Storage Providers

Cloud storage provider is a contracted company that provides cloud-based storage infrastructure, platform or service. Companies usually pay only by amounts of contracted service, depending on usage. As a result, users can rent computing nodes in large commercial clusters through various providers such as: Amazon Web Service, Google Cloud Platform, Microsoft Azure, among others.

2.3.1 Amazon Web Services

AWS is a cloud services platform that offers various services and features to help companies grow. Some services offered by AWS include compute engine, storage, database, blockchain, networking, DevOps, ecommerce, high performance computing, internet of things, machine learning, mobile services, serverless computing and web hosting. Because of this large number of services available, companies like: Spotify, Airbnb, Shazam, Adobe, Siemens, among others, have chosen AWS as a service provider.

The service used at AWS for the tests in this work was the Elastic Compute Cloud (Amazon EC2). EC2 is a service whereby you can create virtual machines and run them on one of Amazon's data centers. You can choose from a variety of machine configurations that have different processing powers, memory configurations, and virtual hard drive sizes (Amazon, 2020). The service also provides full access to computing resources and allows environment settings to be changed.

2.3.2 Google Cloud Platform

GCP consists of a set of physical assets, such as computers and hard disk drives, and virtual resources, such as virtual machines (VMs), which are hosted in Google's data centers around the globe (Google, 2020b). Some services available in GCP are compute engine, storage, migration, Kubernetes engine, databases, networking, cloud spanner, developer tools, management tools, API management, Internet of things.

The service used in this work was Google Compute Engine. Google Compute Engine delivers virtual machines that run on Google's data centers and global fiber networks. Compute Engine tools and workflow are compatible with cloud computing, offering load balancing of scaling from individual instances to global instances (Google, 2020a). Some companies that use GCP services are: HSBC, Twitter, PayPal, Latam Airlines, LG CNS, among others.

As for the cloud providers used in our article, it can be said that the features and tools are similar in both, the list of available services is long and continues to grow. In both providers there is the availability of price calculator. By Providing details about desired services, you can see a price estimate.

2.4 Benchmarking Databases on the Cloud

Database performance evaluation is a recurrent subject in research works addressing new technological advances. Several benchmark tools exist and cloud computing brought even more possibilities to this scenario. Before elaborating our experiments, we reviewed some related works concerning database performance evaluation.

In their studies, Ahmed et al. (Ahmed et al., 2010) and Sul et al. (Sul et al., 2018) present the differences in performance analizing the network, CPU performance, memory and query time of the database, according to the Sysbench parameters. The research of Kasae et al. (Kasae and Oguchi, 2013) was performed in hybrid cloud environments, where they used the Pgbench PostgreSQL benchmark tool for database performance evaluation.

In their work, Guo et. al (Guo et al., 2014) uses the Siege benchmark, which was designed to allow Web developers to evaluate the performance of their code and how it behaves on the Internet. Liu et al. (Liu et al., 2014) however, used the CloudBench benchmark. The analysis tool is a framework that automates the evaluation and benchmarking on cloud scale, through the execution of controlled experiments where full-app workloads are deployed automatically.

Abadi (Abadi, 2009) discusses the limitations and possibilities of deploying data management techniques on emerging cloud computing platforms such as Amazon Web Services. The author also presented some features that a Cloud Data Management System must have when designed for large scale data storage.

To the best of our knowledge, there is a lack of studies focused on the performance of queries over the Moodle database. Though general benchmarks may be useful to assess some metrics, it is also important to consider application-specific queries and tables.

2.5 Benchmarks

In this work we used some tools to compare the performance of the Moodle database in the cloud. The chosen tools were open-source and unrestricted regarding publication of the obtained results. A description of each of the benchmarks follows.

2.5.1 Pgbench

Pgbench is a software made for the execution of benchmark tests on PostgreSQL. It runs the same sequence of SQL commands over and over, possibly in multiple concurrent database sessions, and calculates the average transaction rate (transactions per second) (PostgreSQL, 2020). Transaction is an execution unit that a client application runs in a database. Pgbench is a contribution from PostgreSQL, which performs load tests and assists the analysis of the performance of your Postgres database.

2.5.2 Sysbench

Is a scriptable multi-threaded benchmark tool based on LuaJIT. It is most frequently used for database benchmarks, but can also be used to create arbitrarily complex workloads that do not involve a database server (Kopytov, 2019).

The Sysbench software is designed to measure parameters for a system running a database under intensive load. In addition to database performance, it also allows you to test file I/O performance, scheduler performance, memory allocation and transfer speed and POSIX (Portable Operating System Interface)[1] threads implementation performance.

2.5.3 Moodle Benchmark Plugin

This plugin was developed by (Pannequin, 2016) to be installed in Moodle, in order to perform benchmark tests that report: server speed, processor speed, hard disk speed, database speed and Moodle page loading speed, these tests create temporary test files. After installing the plugin in the Moodle environment, the execution is performed by the superuser through the Moodle interface.

At the end of the run, a report is generated stating a score. This score is the total time in seconds that represents the sum of the tests mentioned above. The higher the value of the tests, the less efficient it is. However, the objective of this work is to evaluate the performance of the Moodle database in the cloud, with this, we only use the values of the tests in the database.

3 METHODOLOGY AND ENVIRONMENT

In this section, we present the methodology used to execute our experiments with Pgbench, Sysbench and the Moodle Benchmark Plugin. The settings used for Amazon and Google instances will be displayed, in addition to our physical computer.

3.1 Methodology

The methodology for performance analysis is described in Database Benchmarking (Scalzo et al., 2007), which recommends restoring the database to its initial state before running a new test, reproducing the tests again, making comparisons with results.

To select which benchmarks and performance analysis tools would be used for the experiments, we carried out a series of small tests on the instances of providers. The intention was to verify the behavior of these tools, their results and its configuration.

Data was imported into Moodle database to be similar to an institution's database. We included registrations of 10 courses and 627 users. The imported data was exactly the same for all instances.

Our settings for running the experiments were the same. We carry out the experiments on different days and times, to ensure that the values have not suffered momentary interference from the instances, but always running the same experiments simultaneously in the instances. In each experiment, we performed five runs and calculated the arithmetic mean to obtain a result.

3.2 Environment

For these experiments two instances were configured in the cloud, one on Amazon EC2 and another on Google Compute Engine, both for free with configurations showed at Table 1. Google does not make clear the CPU model. Thus, a benchmark was used to verify the performance of the CPU (Kopytov, 2019),

[1] POSIX a family of open standards for operating systems.

RAM (Kopytov, 2019) and hard disk (SourceForge, 2018). The performance of the hard drive and the processor in the instance of Google was superior in relation to Amazon, while in the instance of Amazon the RAM executed more transactions per second (TPS).

We run all the benchmarks on our local computer in order to compare Google and Amazon results with a significant baseline. On this physical computer we had full control and made sure there was no additional workload while running the benchmarks. Thus, this scenario is what we consider as a baseline.

Table 1: Configurations of the VMs utilized in the experiments.

Service	AWS	GCP	Local
Zone	South America	South America	Localhost
Processor	Intel(R) Xeon(R) CPU E5-2676 v3 (haswell)	Intel(R) Xeon(R) CPU E5 (broadwell)	Intel(R) Core(TM) i5-8250U
Speed	2.40 GHz	2.20 GHz	1.60 GHz
Cores (per socket)	1	1	4
Threads (per socket)	1	1	2
Hypervisor	Xen	KVM	no
RAM	1 GB	1 GB	8GB
L1 data cache	32 KB	32 KB	32KB
L1 instruction cache	32 KB	32 KB	32 KB
L2 cache	256 KB	256 KB	256 KB
L3 cache	30720 KB	56320 KB	6144 KB
Operation System	Ubuntu 18.04	Ubuntu 18.04	Ubuntu 18.04

4 EXPERIMENTS AND RESULTS

In this section, we present the results obtained in our experiments, executed with the benchmarks Pgbench, Sysbench and the Moodle Benchmark plugin. The results obtained were compared with the results of the tests using the local installation of Moodle.

According to Scheuner et al. (Scheuner et al., 2014), "unfortunately, cloud benchmarking services are complicated and prone to errors". Based on this, we opted to perform the same tests with three benchmarks in both instances, during random days and times. The same test was executed five times in each transaction / client, and the arithmetic mean of the tests was calculated. In each new test, reset the tables used by Pgbench and Sysbench, each one composing the parameters in an empirical way.

The experiments were performed focusing on the goal of measuring only the performance of the database. All test scripts and documented commands are available at: https://github.com/karinawie/PAD-UFSM.

4.1 Results with Pgbench

Pgbench performs load tests on the database involving five clauses for each executed transaction, SELECT, UPDATE and INSERT, thus analyzing the bank's performance. Before running Pgbench, it is necessary to make settings where Pgbench requires specific tables for the execution: pgbench_accounts, pgbench_branches, pgbench_history and pgbench_tellers.

In the graphs presented in Figures 1 and 2, the vertical axis shows the values of TPS and the horizontal axis the variations according to the number of customers. The Pgbench experiments were executed five times for every 20 clients, then we doubled the number of clients and executed the same five executions until reaching 280 clients, that is, we started with 20 clients after 40, 80, 160 and finished with 280, in total twenty-five runs for each instance using Pgbench. At each rerun of the scripts, the base has been reset.

In the Pgbench results, we present the arithmetic mean of the five executions. The duration of the test was measured in TPS. The higher the result value, the more efficient the performance. It is possible to see in the graphs that in both experiments Amazon was more efficient in all results compared to the Google instance. Compared to our baseline, the performance of our instances in the cloud is lower, but we need to take into account the hardware configurations.

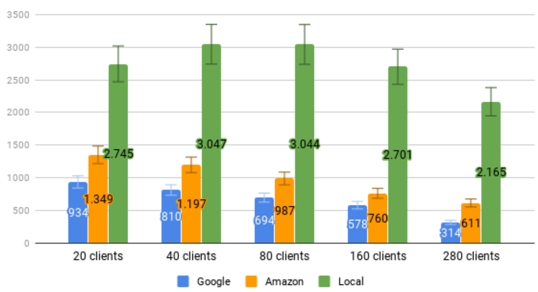

Figure 1: TPS values in the instances.

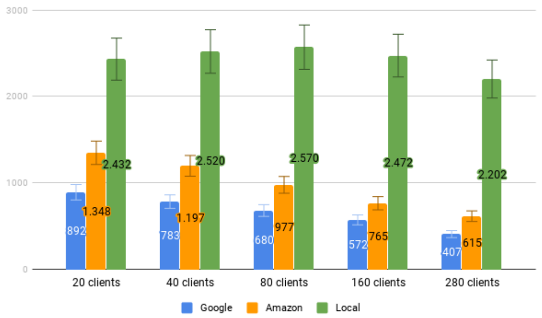

Figure 2: TPS values between the instances.

4.2 Results with Sysbench

Sysbench executes a specified number of threads and they all execute requests in parallel. We started the tests with 20 threads, then doubled the value to 40, 80 until we reached 160 threads. The last execution had to be finalized in 160 threads, because from this value the instances in cloud present difficulties in fulfilling the request.

On each run, the test database was reset and the tests were performed again. We repeated the same command five times by changing only the value of the thread. Finally, we averaged the TPS. Figures 3 and 4 show the results and the average response time variation of the experiment. Amazon's performance was superior compared to Google's, according to the settings we use.

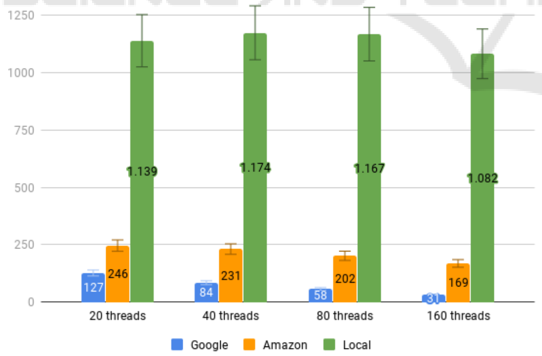

Figure 3: Divergent values throughout the test between the instances.

4.3 Results with Moodle Benchmark Plugin

The benchmark plugin was installed on each Moodle environment in both instances and local. As the objective of this work is to evaluate the performance of the Moodle database in the cloud, we use only the items related to the performance in the database, which are items numbered 6, 7 and 8. Item 6 inserts 25 courses temporarily. Item 7 performs the same selection one hundred times and item 8 performs the same selection 250 times, but different from item 7.

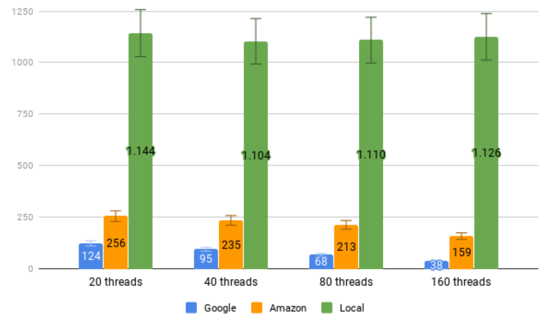

Figure 4: Divergent values throughout the test between the instances.

The experiments had the duration of their executions counted in seconds. The lower the average time in seconds, the better the provider's performance. We add the results of the 3 items (6, 7 and 8) in the five executions of each instance, then we calculate the arithmetic mean.

Figure 5 shows the results of experiments with the plugin. Again, compared to instances in the cloud, the Amazon instance performed better than the Google instance.

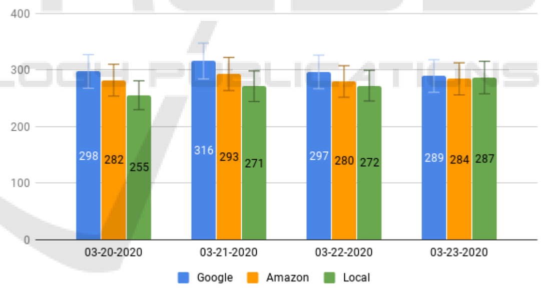

Figure 5: Experiment duration in seconds.

5 CONCLUSIONS

This article presented a preliminary performance investigation of the Moodle database in cloud environments. In our experiments, we used the free version of the cloud providers AWS and GCP. We installed Moodle in instances on both providers. We used the Pgbench and Sysbench database benchmarks and the Moodle Benchmark Plugin to gather performance metrics. We run all the benchmarks on our local computer in order to compare results with a significant baseline.

According to the data we gathered from the benchmarks in the environments we considered, it is possible to observe that Amazon instance presented better performance in all cases, compared with instance Google. Due to the inherent characteristics of cloud environments, this results may change within certain limits when we chose different service/billing plans, so this remain to be investigated. Also, it may be useful to gather more performance metrics from specific cases of Moodle usage, according to the characteristics of each educational institution. To encourage reproductibility and further work, we made our data and scripts available in a public repository.

Our choice of a IaaS service model granted us autonomy to install different benchmarks and to perform measurements directly from our instances. This may not be possible when Moodle is provided in a SaaS model, but such alternative still deserves to be investigated. Obviously, performance is not the only factor when deciding to migrate Moodle to the cloud, but simplicity and cost effectiveness are useless when we are not sure how the system will perform under a cloud environment.

As a suggestion of future work, it is intended to conduct experiments with other cloud providers not contemplated in the study, use other benchmarks for performance evaluation. Another important aspect is the possibility, if there are financial resources, to perform the cost-benefit comparison on a paid way. For the customer, may need to hire a service that performs best, but that fits within your budget.

REFERENCES

Abadi, D. (2009). Data management in the cloud: Limitations and opportunities. *IEEE Data Eng. Bull.*, 32:3–12.

Ahmed, M., Uddin, M. M., Azad, M. S., and Haseeb, S. (2010). MySQL performance analysis on a limited resource server: Fedora vs. Ubuntu Linux. In *Proceedings of the 2010 Spring Simulation Multiconference*, SpringSim '10, pages 99:1–99:7, San Diego, CA, USA. Society for Computer Simulation International.

Amazon (2013). AWS marketplace: Moodle cloud. Accessed: 2018-09-26.

Amazon (2020). Amazon EC2. Accessed: 2020-02-14.

Bitnami (2020). Moodle Cloud Hosting, Deploy Moodle. Accessed: 2020-02-16.

Caminero, A. C., Hernandez, R., Ros, S., Robles-Gómez, A., and Tobarra, L. (2013). Choosing the right LMS: A performance evaluation of three open-source LMS. In *2013 IEEE Global Engineering Education Conference (EDUCON)*, pages 287–294.

Google (2020a). Google Compute Engine documentation. Accessed: 2020-02-14.

Google (2020b). Why use Google Cloud Platform? Accessed: 2018-02-14.

Guo, X., Shi, Q., and Zhang, D. (2014). A study on Moodle virtual cluster in cloud computing. In *2013 Seventh International Conference on Internet Computing for Engineering and Science(ICICSE)*, volume 00, pages 15–20.

Jamsa, K. (2013). *Cloud Computing: SaaS, PaaS, IaaS, Virtualization, Business Models, Mobile, Security and More*. Burlington, MA : Jones & Bartlett Learning.

Kasae, Y. and Oguchi, M. (2013). Proposal for an optimal job allocation method for data-intensive applications based on multiple costs balancing in a hybrid cloud environment. In *Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication*, ICUIMC '13, pages 5:1–5:8, New York, NY, USA. ACM.

Kopytov, A. (2019). Sysbench manual.

Liu, X. X., Qiu, J., and Zhang, J. M. (2014). High availability benchmarking for cloud management infrastructure. In *2014 International Conference on Service Sciences*, pages 163–168.

Mahmood, Z. and Hill, R. (2011). *Cloud Computing for Enterprise Architectures*.

Mell, P. and Grance, T. (2011). The NIST definition of cloud computing. Accessed: 2018-10-12.

Miller, M. (2008). *Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online*. Que Publishing Company, 1 edition.

Moodle (2016). Arguments in favour of PostgreSQL. Accessed: 2020-01-12.

Moodle (2019). Moodle hosting from the people that make Moodle. Accessed: 2019-09-26.

Pannequin, M. (2016). Moodle benchmark. Accessed: 2018-10-12.

PostgreSQL (2020). PostgreSQL documentation: devel pgbench. Accessed: 2020-02-16.

Scalzo, B., Kline, K., Fernandez, C., Ault, M., and Burleson, D. (2007). *Database Benchmarking: Practical Methods for Oracle & SQL Server (IT In-Focus series)*. Rampant Techpress; Pap-Cdr edition.

Scheuner, J., Leitner, P., Cito, J., and Gall, H. (2014). Cloud work bench – Infrastructure-as-Code based cloud benchmarking. In *2014 IEEE 6th International Conference on Cloud Computing Technology and Science*, pages 246–253.

SourceForge (2018). Hdparm. Accessed: 2018-10-10.

Sul, W., Yeom, H. Y., and Jung, H. (2018). Towards sustainable high-performance transaction processing in cloud-based DBMS. *Cluster Computing*.