Cloud Computing Trends in Higher Education: Cost and Necessity Modeling

Daniel Kruze

CSCI-49500: Explorations in Computer Science

With Faculty Advisor Ms. Kelly Van Busum, Instructor Dr. Qin Hu

Spring 2023

ABSTRACT

Within the last decade, cloud computing has come to supersede the traditional server model for essentially every application of large-scale data storage and software distribution. Technologies like AWS, Google Drive, Microsoft 365, and even more novel things like Github Kraken, Replit, or ChatGPT have proven beyond a shadow of a doubt that the use of cloud computing systems is not only efficient but lucrative, and many companies the world over have come to agree.

The question for educators has become how cloud computing can be used in the classroom. The purpose of this document and its supplementary material is to provide an educated answer for the higher learning environment, specifically in computer science (but also in general,) as to how beneficial it truly is to migrate existing computer systems to cloud computing systems, for more than simple data storage.

Cloud computing systems are distinct from traditional server systems in that the components of a system are not necessarily hosted in one place on one server, and certainly not on any local installation. The idea is that a group of servers is connected to run entire environments on a server-side basis, allowing clients (users) to access entire software packages at will without hosting or installing anything locally. For this reason, it has been dubbed “software as a service,” and a non-insignificant amount of universities have already begun to adopt cloud technologies for software distribution to prevent complicated rollouts of local software that must be connected via proxy to a central server.

As many IUPUI students and staff could testify, however, there are simply many more battles to win with the migration. Such is the case for every university in the country, and the purpose of this paper in a more *detailed* sense is to provide a data-driven, empirical argument for or against the migration of existing systems to cloud systems, based on a set of criteria. These will include short-term cost, long-term cost, ease-of-use, necessity, and of course capability, i.e. “should the average university pay for cloud servers, or do any of them have access to existing ones?” The purpose of this document is *not* to make a case in either direction for the technologies in the consumer market, nor is it to analyze the structure of the technologies themselves. It will focus mainly on case-by-case analysis of existing migrations and data models for the future effectiveness of cloud computing technologies.

INTRODUCTION

Higher education has been more or less entirely subsistent on computing for a number of years now, depending in every capacity on deliverables via online services to and from students to conduct grading and testing. In computer science, the original move from flash-drives and discs to repositories and emails was a simple one—it is no longer so. The demands of the modern computer science classroom gave sophisticated in order to keep pace with similar demands from the tech employment industry, and as such it has become necessary for educators and financiers alike to pay attention to the way student data is organized and stored.

Enter cloud computing services: Instructure Canvas, Google Docs, and Microsoft Onedrive are the collective tip of the spear. As a student at one of 4000 universities [13] nationwide, one might use Canvas every single day to view upcoming assignments and turn in completed ones, and likewise professors might use it just as often to check in on messages or assign new deliverables. The underbelly of Canvas, i.e. the system on which it is built and hosted, is a cloud computing system—a system in which multiple servers are linked together to not only store data from a swath of users, but provide all those users with a server-side environment for viewing and manipulating existing data, or interacting with other users. All the services that canvas provides are hosted online and accessed via web browsers, including (but not limited to) submitted assignments, files germane to specific classes, emails between students or between students and professors, and schedules for upcoming classes or previous classes.

At the risk of sounding like an advertisement, it might be simple to say that a cloud computing service capable of organizing all of this in one spot is an undeniable boon for universities, and naturally this specific service—Canvas—is much simpler than most alternatives. However, there are many more angles to consider before determining whether a cloud computing system is an effective tool to implement for a classroom, and the discussion of those angles will be oriented around computer science classrooms.

Primarily, the subject of this paper will be that of the cost of a cloud computing system; how much does it cost, in terms of time and capital, to integrate an educational environment with a cloud computing system? Additionally, the security of a cloud computing system must be considered immediately after a cost is figured, because not only is academic integrity a top concern in a classroom environment but a classroom in a university could lend itself to some sensitive unpublished material, either academic or personal in nature. Not only that, but there is a certain degree of unnecessary headache involved with a cloud computing system, in that not every environment *needs* to be integrated into such an environment if the existing one is proven to be successful; this will be addressed as well.

The majority of the cost discussion will concern the exploration of existing models in the field for predicting price under specific circumstances. In tandem with these explorations, some models will be provided (listed in the **figures** section) that represent potential trends in cloud computing systems, based on the current implementations. Those implementations will be referenced when relevant.

THE MIGRATION PROBLEM

The primary conceit of the cloud-computing system's implementation into a university environment is that existing systems would need to be migrated into it. In some cases, this could be a migration from a traditional server system into a cloud computing system (such as an email system being migrated into a cloud environment, like over 60% of universities in America [10] have likely already done,) or from one cloud computing system to another. For the sake of argument, both ought to be considered in their own right, although for some services it is *more likely* the latter, not the former.

Migrating a traditional system to a cloud system could prove to be a very costly option for a university not currently allocating funds to a cloud system. In a traditional server, data is hosted by a server while software for accessing and interpreting the data is hosted on a client machine; note that in this situation, the client would be a student. This means that if every student in a given class is expected to use the same software to access a database hosted on a traditional university server, each student must individually install and configure a technology on their own computer to access the data, and then their individual environment must somehow be gauged as secure enough to access the hosted data without altering or damaging it. Understandably, this can be disastrous and costly—in fact, universities have long been considered to be "lagging behind" [6] other sectors in adoption of clouds largely because the traditional method is so complicated, it's perceived as though there's little reason to adopt a new system that might not be easier either way. These concerns (besides the cost concern, which will be evaluated later,) actually make up the bulk of proposed issues with moving traditional servers to cloud servers, with many universities taking issue with the reliability and security [6] concerns that come with trusting a system to be run online by a third party, not to mention the way that system will be perceived [6] by a student and staff body.

These are legitimate concerns, for which cloud computing systems for the educational environment have developed methods to address. However, as promised, there is also concern for similar issues when migrating from an *existing* cloud computing system to a new one. After all, there are about 3000 websites in the United States that use the Instructure cloud alone [9] to power themselves, which may seem insignificant compared to the whole of websites on the internet, but is not insignificant when considering how many universities exist in the United States. Many universities have already come to use clouds for the exact purposes detailed in this paper's introduction, and as popular as Instructure's Canvas service is, it is far from the only cloud system on the market. The general motivation behind migrating from one cloud to another would be for cost-effectiveness (a nebulous term that will be thoroughly defined in the following discussion,) but other motivators for switching between clouds include lock-in concerns, privacy concerns, [6] and even by popularity in some cases. [7]

For both of these situations, the cost is essentially the same. The system migrated from makes little difference in the monetary impact of adopting a new system, and the amount of labor-hours necessary to transfer data from either a traditional server or an existing cloud into a new cloud is not reasonably quantifiable, or is otherwise outside the scope of this paper. When the "cost," then, is discussed for migration into a cloud computing system, it is measured in two ways: the cost in computing terms and the cost in monetary terms. More specifically, the "cost" of a migration is given as the capital required to acquire and implement a given system, and the stress on integrated systems or networks it would cost to use.

For the latter concern, existing models are difficult and often highly specific. A model needs to be constructed for a particular architecture and then generalized from that architecture, then tested against other specific architectures to be trained. [4] While these general models do exist, [4] they are fraught with limitations, in that cloud computing is a relatively novel technology. Few long-term implementations of cloud computing systems exist for which there is also a statistically relevant quantity of specific data, and downtime needs to be minimized during migration in order to provide reliable data. [4] Naturally, these factors make it difficult to predict how much stress a cloud computing system can incur, both for itself and for integrated, existing systems; however, models do exist that are proven to be reliable for many large-scale architectures. Specifically, the given models are accurate up to approximately 200GB of data, and were tested using web-based cloud architectures. [4]

Such models can be used to extrapolate costs for migrating architectures, and typically result in combinations of linear models for network consumption alongside non-linear models for installation cost, online and offline. For example, the model given by [4] for predicting migration cost based on the summation between adding bandwidth models to installation models for individual system components (the exact formula which forms its base is given in section VI, [4]) was implemented using data supplied by the Azure system, wherein the model produced predictions for best and worst infrastructure costs as the size of the system increased. The general result trended upward in a nearly linear fashion as one might expect, however the impact of the migration sequence—the order in which component systems are transposed into a cloud environment—seemed to be larger than any other parameter. [4]

This implies that while migration cost, in computer memory that is, *can* be modeled effectively for some systems, the efficacy of a model seems to rely more on the subjective details of a given environment, rather than any general process.

Of course, this seems obvious—a general model can only provide so much information about a potential migration if the model is built with a specific target system in mind, vis-à-vis the system being migrated away from. What, then, of the monetary cost? For a university, it's typically more relevant to consider the financial stress of a system migration being that most universities outsource the management, construction, and optimization of server systems to third parties. This subject, naturally, has more competing models for cost based on a swath of factors, such as the hosting of the system, the market cost of the software, the valuation of the company, and so on.

The first model worth considering is given by [11], and is built in lieu of so-called "value based pricing," and focuses on modeling the given value of cloud computing software by extrinsic, non-marketable features that are typically not considered, leading to costs for software alone being higher in the long-term than they are projected by other models. The model is trained using primarily AWS (Amazon Web Service, a very popular cloud system for large databases, among other things, [8]) data and was created to show that the average consumer is paying for more than was previously expected for cloud expenses *after* adoption of a cloud computing system, and that the general price of cloud computing systems may be decreasing over time. [11] This model is built on a form of logistic regression equation that includes an additional factor for "marginal costs," which in this context refers generally to the flat cost of expansions or updates to a given system. [11]

Whereas the previous model for infrastructure accepted parameters pertaining to continuous data, such as processor speed or bandwidth, [4] this model's parameters are only partly continuous (price) parameters, and are for the large part resultants from individual functions to calculate markup prices from given markets. [11] While the authors have admitted that it can be difficult to specify the value of each parameter for a general model, markets with significant data have allowed the model to be refined and rewritten for individual case studies as a ”hedonic” logistic regression model, including a natural logarithm, which is detailed in section 4.2 of [11]. Testing of the model using, again, AWS data revealed that different packages for AWS software have relatively similar performance and price points, and that a more statistically significant factor for the price prediction of a specific cloud package has to do with a *time* factor, given in [11] by a “time dummy” variable. To avoid getting ahead of the general conclusions of this paper, the big takeaway from this model is that the price for adopting a particular instance of a cloud computing system regardless of the conditions of the existing system, *can* be modeled qua specific market data, and that the given models for doing so predict that the time of adoption and time spent migrating or managing a system have a larger impact on cost than other factors [11].

There are, however, other models that are trained less significantly on specific data, rather for specific kinds of cloud computing systems. How could a model be built to account for clouds operated by institutions themselves *and* clouds operated by third parties? So-called “federated clouds” exist that consist of systems where the twain meet, and the model given by [12] subsists on their implementation. Rather than build a model with highly specified parameters, it can be assumed that a well-constructed model for a federated system could be redefined to apply to either a provider or a client-operated cloud computing system.

The model is actually given as a *series* of models designed to handle a multitude of pricing situations: consumption, subscription, advertising, market-based, and group-buying (these are defined in section 3 of [12].) For each pricing agreement, a linear model is built, parameterized by the volume of data stored and transferred, and the methods by which that data is managed. The idea behind making these multiple models is to apply them selectively to a system based on its particular package specifications, so that a price can be gauged for a potential migration based on the specific needs of the business. [12] Experimentation shows that various implementations of the models using two existing data management technologies indicated that the cost of a cloud computing system in general may be heavily influenced by qualitative factors beyond the simple “provider/client dichotomy introduced earlier in this paper, such as the number of jobs sent through a system or the reputations of the companies on either side of a system purchase. [12]

From both of these monetary models [11] [12] and the given infrastructure model, [4] it can be reasoned that a migration is a highly complex issue, if nothing else. A specific price for a migration can only be reasonably evaluated for specific use cases and with a well thought-out time frame in mind, not to mention the cost can very greatly before the migration (dependent on whether the system is operated by the provider or the client,) and after the migration (dependent on the existing systems being migrated away from, and the time frame of the migration itself.) A migration from an existing cloud computing system can be less taxing on infrastructure, but may not necessarily have any impact on the monetary cost of adopting a new cloud computing system. Migrating from a traditional server system into a cloud computing environment is presumably about as complex and expensive as it sounds on paper.

AVENUES FOR IMPLEMENTATION

What, then, are all of these models useful for? It has become clear that there are ways to predict the cost of migrating to a cloud computing system, and that there are more concerns than just cost alone when considering adopting a cloud computing service. What can a cloud computing system actually *do* for a university that would make it worth confronting these models in the first place? More specifically, where and how can a cloud computing system be implemented into a computer science classroom? These questions are more the concern of educators themselves, whom are the target of this portion of the exploration, being that they are the ones responsible for teaching with the systems. While the cost of a system is more relevant to those who control budgets and curricula, there is obviously a certain amount of motivation behind any migration qua the actual usage of the system.

Before discussing the ideas of computer science specifically, however, it is important for educators to understand how a cloud could be generally utilized in *any* kind of classroom. What advantages do clouds have vis-à-vis the logistics of the classroom, above the specifics of a curriculum? Consider the challenges involved with organizing data for a university class: a class could have upwards of 100 students yet only one instructor who is responsible for mandating the technologies the class will use, and collecting (then evaluating) deliverables from each student. Not only that, but these classes in some curricula could have deliverables multiple times a week, on top of exams and in-class activities when they are necessary—how could a cloud alter this model?

For starters, many curricula involving “quantitative methods [5]” such as mathematics, statistics, or technical writing are curricula that are interdisciplinary, meaning students use the skills from one curriculum in other curricula. These quantitative methods have curricula typically built around multitudes of competing methods and algorithms, as well as formulaic techniques, which in a modern classroom can typically involve a decent amount of difficulty and even anxiety with deliverables on part of the students, [1] [5] being that it can be confusing to find ways to effectively implement the multitudinous methods involved using various technologies—this typically manifests as web-based services, but there are other, less cost-effective options as well.

One such option, which is only less cost-effective than others because it is simply not free, would be any of Google's cloud computing systems, in particular Sheets and Docs. These are clouds in that the data that can be uploaded from a student machine and then manipulated freely using services and programs hosted by Google and accessed via Google's interface and Google's browser, then retrieved at will via a network from cached preferences. For any curriculum involving quantitative methods, the built-in statistical analysis features of Google Sheets, which include support for packages and plug-ins, could create an environment for students where all of their deliverables could be created, updated, and uploaded into one system. From there, the data could be accessed by students individually and by an instructor at any given time via separate Google accounts, [5] and less quantitative deliverables could follow the same process via Google Docs. Furthermore, there is no need to even speculate as to the ease-of-use of such a system, because linear models of student evaluations for use of Google Sheets in classrooms revealed that with exposure to the system over time, student's reported feeling more positive about the systems and about their own efficiency qua the deliverables [5].

This generalized example sets up the discussion moving forward: what cloud computing systems are really applicable to universities? What could a professor reasonably acquire, install, and teach to a class of computing students that would be relevant in any way? Obviously, Google's cloud technologies and Instructure's cloud services have already been examined (and implemented [8] [9], as will be seen,) but there are also services like Github, AWS, and Microsoft Onedrive that could be germane to universities' needs. Onedrive, like Instructure's Canvas, [13] is already implemented by the very institution that this paper pertains to, so discussion of those services would be redundant. Google Sheets, as previously examined, already has an obvious avenue for implementation, making its examination also redundant; thus, the remaining discussion in this section will be oriented toward Github, being that AWS is not, by itself, necessarily useful in most classrooms given its tendency toward large-scale business implementation.

Take, for example, a curriculum for data science and statistics, which are the primary foci of this paper. In a university environment, these classes are typically conducted using tools such as IDEs, data analysis technologies, and hosts of programming languages such as R or Python. It can be very difficult, as a professor, to consolidate submissions from students using a large variety of local builds and technologies, as we have seen. [5] This is especially true considering it may require the personal installation and usage of each and every corresponding technology on behalf of that professor to actually provide feedback for deliverables. A typical solution for this is to simply mandate that students all download and use the same software package on their own systems, which is—pejoratively—a complete crapshoot, being that there is no realistic way to guarantee that every student can install the software successfully or even install the correct version, those concerns standing apart from the obvious problem of required technologies that aren't open-source and require licenses.

Could a cloud computing system provide respite from these problems? Indeed, many professors at IUPUI already require the use of Github to host data pertaining to builds of deliverables and provide an environment for students to build programs via a server. How could a cloud improve this model?

The implication of Github is germane to this consideration: Github represents a cloud model, not simply a version control model, and is thus worth considering as an example of a cloud to be implemented in this situation. Github includes not only a way to store and access data via server, but also a way to edit code, write code from scratch, collaborate in real time with others, and even query an AI to have code written or revised on the spot. A curriculum involving statistics-driven code—which is certainly a "quantitative method, [5]"—could be facilitated with Github to make submissions easily visible to instructors, store data for students, allow or disallow collaboration for labs, and even synchronized installations for coding languages and environments. [3] Practically, Github provides a suite of remotely-hosted systems and services that can coordinate software installations and deliverables for instructors, and provide storage and support for students. In fact, citation [3] provides some details of a particular implementation of Github into a STAT 184 classroom at Penn State, focused on the R programming language, which included the use of Github (for deliverables) and Github Classroom, which is one of Github's services that instructions, assignments, and technologies to be hosted *and* assigned by instructors to students. Additional examples include a STA 210 statistics course at Duke University, as well as a graduate course at Brown University in biostatistics, [3] both of which implement the aforementioned cloud technologies to give students more control over their own work and educators less stress over the installation of necessary software.

These cases simply go to show that a cloud service such as Github is entirely suited to solving some of the problems associated with computer science classrooms qua logistics, as such cloud services have already been implemented before. [3] Of course, for simple organization there are other less complex services available [1] that may be better suited to courses with less focus directly on software, i.e. mathematics or introductory statistics. This isn't to say that classrooms outside of computer science couldn't benefit from these systems, however, being that these interdisciplinary methods that were determined earlier could provide new advantages for students in other curricula to use computing to their advantage. According to [2], reports from environmental science students within the last decade indicate a trend toward computational illiteracy that has driven many universities toward data science requirements, being that modern statistics relies on computation so significantly. It is worth noting that in such a situation, it would be remarkably useful for an instructor to rely on a cloud computing system to facilitate computer education in an environment where many students, regardless of background, may indeed be new to coding. Likewise, [2] recommends the use of RStudio and RStudio Cloud, as well as the use of Github for RMarkdown documentation to teach the R statistical language in such classrooms.

What can be observed in all of these cases is that cloud computing systems can be implemented in any STEM or STEM-adjacent classroom to noticeable advantage, being that they provide convenient ways for instructors to coordinate software with students, and for students to store and access data at will without sacrificing local space.

THEORETICAL IMPLICATIONS

So, with an idea of the cost and a host of different use cases to consider, the question moving forward for institutions is this: is the cost of a cloud computing migration worth its weight in educational benefit? The question now is more for those who set budgets and curricula than for professors, of course—as simple as it is to imagine that any coding class could be streamlined by quick and simple access to powerful data manipulation tools, it's much *less* simple to imagine that a university could simply use “x” rather than “y” for its software packages with no consequences.

Consider the models for cost and the implementations that have already been discussed in some detail. It's already obvious that a cloud computing system can, in practice, alleviate the stress of tech illiteracy and inconsistency in both computer science disciplines and in other disciplines as well. [2] Plenty of the services a cloud provides can (and, in fact, already do, [9]) provide a unique advantage to every party involved in a classroom, allowing software rollouts to be virtually abolished in favor of simple, browser-accessible technology with built-in database technologies, and it's no secret that this is significantly faster for students, if nobody else, than a traditional server setup. However, not all of these services are so obviously beneficial and many of them have not been largely implemented yet, with advanced cloud computing systems that include functionality for meetings, messaging, or even entire, separate applications only being implemented in somewhere under 30% of US universities. [10] Is this because they're too costly to migrate to? It may well be—research compiled in [6] indicates that the cost of a cloud computing system is among the top 10 concerns for a university when considering migration, and despite the fact that the *same* research indicates that a migration could actually *save* money as well as processing power, per the model discussed earlier, [11] not every university may be aware of this due to the novelty of the technology.

However, that same research [6] *also* indicates that higher priority interests than just cost may exist: the flexibility of an environment and the ability for that environment to be portable are growing to be very large concerns for potential migrations as time progresses, not to mention the ability of a system to be portable and reliably accessible. Could cloud computing systems exist to fulfill these needs, or is this a barrier to their potential usefulness? For one answer, consider the Google cloud that has been mentioned previously. [5] The google cloud environments are completely mobile-friendly and are already highly portable, and rely on technology already so common that no student would reasonably hesitate to adopt it. This example does, however, raise the previous concern of cost quite significantly, as Google's technology is far from free—the monetary cost of Google's cloud computing systems can be quite taxing for institutions being that Google handles most processing on their end and allows access to their clouds via browsers. This kind of technology is obviously stressful for Google's servers, and beyond just the cost of running the system itself, the use of it as a university requires providing Gmail accounts for all given students and requiring them to manage their own licensing and security.

This tradeoff is largely up to the discretion of the university, as not all cloud computing systems follow the same practices. No one cloud is necessarily perfect for every single use case, and essentially every popular technology has anecdotal quirks that a university might consider when choosing to implement them, but there is a sort of objective through-line that defines cloud implementation. That line would be the notion that the questions posed earlier by the models analyzed for cost can, indeed, be answered as predicted: the cost of a system migration is largely dependent on existing architecture, but is generally favorable in the long-term for saving processing speed and money. Additionally, the usefulness of a cloud computing system is likely pretty significant, given that many existing clouds already provide functional answers to many of the concerns that universities have when migrating in the first place.

In summary, based on what has been analyzed and aggregated from existing writings and test cases, it would appear from a distance that a cloud computing system **could indeed** be beneficial to a computer science classroom invariably, if the cost incurred is not too large. While it can be taxing for large system to be migrated, many modern cloud services are hosted through browsers and set up very simply, with much of the back-end work handled by the company that holds the actual cloud, not the university. Furthermore, the monetary cost of a cloud subscription is practically no more expensive than the cost of maintaining traditional servers or paying for licensing for traditional software rollouts, that are also commonly complicated and fraught with technical issues for students. As of 2016, [10] it was reported that almost 80% of the average university students in America were being tasked to use cloud computing systems via browsers *already*, and that does **not** apply specifically to fields where the advantages of clouds can be implemented beyond simple data storage and software rollout. For any computer science department, then, that can afford to maintain traditional servers already, a cloud computing system can really only bring long-term benefits to ease-of-access for students and professors both.

VIABILITY

All things considered up to this point, the big questions have been “answered.” It is safe to *assume* that a cloud computing system could be a cost-effective migration for a university’s current server setup, and it could provide great benefits to students in terms of actual implementation of cloud applications. However, this answer would be informed primarily by speculation and historical precedent—why not analyze trends? If genuine predictions could be made as to the adoption of cloud computing systems in the future, it would allow conclusions to be drawn about the usage of clouds in the classroom that might not become entirely irrelevant when new, novel technologies (such as AI) become the new standards for education. Consider a conclusive analysis about clouds including only what has been analyzed up to this point; one could stop halfway through and imagine that clouds are worth the cost because of the headache they save for students and educators, the idea that migration is inevitable due to novelty, and the consideration that traditional server systems are almost equally costly for universities to maintain. The obvious flaw in these assumptions is that cloud computing systems also need to be maintained and updated, and if such systems are phased out by newer more efficient models within the next decade or so, they would inarguably become significant losses if only *some* institutions actually adopted them.

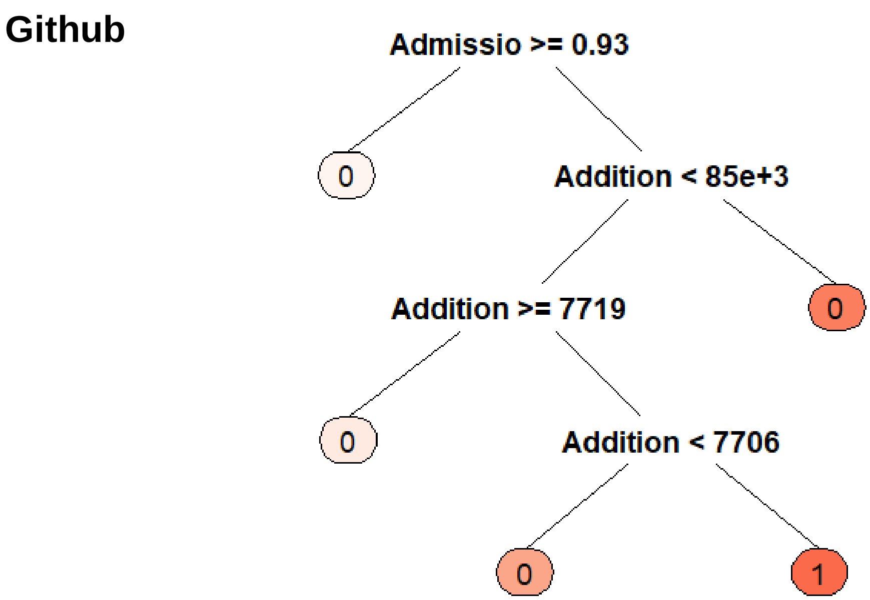
This all goes to say: predicting trends in cloud computing system adoption is a necessary caution for an institution before making the decision to migrate, because the cost of tying sensitive data and complex curricula to a system that may be difficult and expensive to use or maintain could be *enormous* if it needs to be replaced so soon as it was installed. Not to mention the notion that not every institution may even *need* a system with the capabilities of a cloud computing system—for the sake of argument, we will assume for these analyses that the data used pertains to universities with departments robust enough to necessitate the use of such a system, particularly in a computer science department.

For the sake of argument, the data considered for prediction will be for recent usage rates of three of the cloud computing systems most heavily relied upon for examples in this paper—Github, Google Drive, and Microsoft Outlook. These choices should make it easy to contextualize, after prediction, the potential implications of the data beyond simply a “yes these are being employed” or a “no, nobody is using these.” Bear in mind that all of these services offer a large swath of different kinds of cloud services (Google alone offering over 190 different services, [14]) but the consideration is for their most general clients that are typically for operations like messaging, source control, data storage, or webapp access. These are the particular kinds of clients that universities would expect students to learn.

To make predictions on how the usage is expanding for these particular technologies, the idea is to make three collections (sets) of data, one for each technology, and use adoption rates aggregated from [14] and general information supplied by [16] to generate attributes, including a binary “yes” or “no” for whether or not a university (an instance in a set) has adopted this cloud technology. For example, a data set for Github would include around 6,600 instances for every university in the united states as of 2020, [15] and as of that year around 555 universities reported using Github. [14] So, for each university, there will be an ID, some measures of tuition, enrollment, and degree offers, the third of which would obviously have to be an ordinally encoded variable. Therefore, that table will consist of 6,600 universities and their information, 555 of which have adopted Github (with a 1,) and 6,045 of which have not (with a 0.) From these, 75% of the instances will be used to train a decision tree to select universities that may adopt Github, and the remaining 25% will be used to test this model, the accuracy of which will represent, as a whole, the adoption rate by percentage for Github around 2020. Given that this rate will obviously only increase in the future, if the accuracy is sufficiently high enough, then the predictive power of this models could be considered a good indication that migration is worthwhile, or *not* worthwhile if the model has poor results.

The given decision tree will be generated with a repeated CV resampling algorithm using the R scripting language on an RStudio 2022 environment. The functions for handling this algorithm are a part of the “Rpart.plot” library, and the data sets are aggregated from .csv data files supplied by [15] and result sets from [14]. The resultant trees will have predicted results recorded, with a reference image, and the scripts to generate them will be included in a .zip file alongside this report (please look for the “ExplorationsCC.zip” file.)

That reference image is below, with elaboration to follow:



The above graphical representation of the weighted traversal of binary variables in the Github data represents the modus operandi of the decision tree model for these given data sets. Each data set contains attributes for an institutions admissions, location, degrees offered, and collected extraneous fees, all of which were fed into a logistic regression algorithm repeatedly using the aforementioned resampling method. After 10 rounds of resampling in 3 repetitions, a decision tree is generated, and the following graph indicates that the calculated weights for the collected additional fees and the admission rate seem to bear the most impact on the adoption use of cloud computing technologies, or at least for Github, by showing that the biggest splits between adoptive and non-adoptive institutions seem to occur contemporaneously with splits of those attributes.

Similar graphs with slightly different numeric values exist for the other two data sets, which contain more instances of institutions that have actually adopted both Google and Microsoft cloud computing systems. Do these models have anything relevant to say, then? Are these weighted attributes good barometers for measuring adoption of cloud computing systems?

For the tree that generated the above graph, the accuracy of prediction was reportedly 92% according to a confusion matrix, which is a statistical analysis tool as part of the caret library in R, when compared to the actual adoption of those systems in the test data. For the other two technologies—Google Drive and Microsoft Outlook—reported accuracies were 85% and 77% respectively. Specifically, this means that when more institutions have actually adopted a technology, the model struggles a little more to differentiate according to the given parameters between whether another university will or will not adopt that same technology.

```
               Accuracy : 0.9149
                 95% CI : (0.9005, 0.9279)
    No Information Rate : 0.9149
    P-Value [Acc > NIR] : 0.5223
```

```
               Accuracy : 0.8532
                 95% CI : (0.8353, 0.8699)
    No Information Rate : 0.8532
    P-Value [Acc > NIR] : 0.517
```

```
               Accuracy : 0.7717
                 95% CI : (0.7508, 0.7917)
    No Information Rate : 0.7717
    P-Value [Acc > NIR] : 0.5137
```

The above screenshots of the RStudio console represent snapshots of the confusionMatrix() readout for the Github, Google Drive, and Microsoft Outlook decision tree models respectively, as they are applied with predict() to the test data.

This is obvious, however, and the larger takeaway from these models is the conclusion that they all had predictive accuracy higher than 75%, which is satisfactory at least and promising at best given the large sample size of every single accredited university in America. What this all serves to illustrate is that by current metrics, it *may* be accurate to assume that the migration into cloud computing systems by universities in the future can be predicted by current trends, and seems from a distance that the rate of continued adoption can be reasonably assumed to be increasing steadily, and almost certainly assumed in the case of Github. It should be conceited that the actual data for adoption from [14] was not necessarily all public, meaning that the metrics by which the given trees judged adoption are likely not the best metrics to choose—however, considering the data from [15] originally had nearly 3000 attributes, those with access to sensitive data of that nature could use the exact same algorithm to generate even more accurate results.

Conclusively, the results of these predictions indicate that in no specific capacity, cloud computing system adoption is not going to see a decreasing rate moving forward, and current trends in data indicate that it is very simple to predict the adoption of specific cloud computing systems for universities that haven't migrated from other systems or traditional servers.

CONCLUSIONS

What, then, can be gleaned in the end? The migration problem posed at the beginning of this paper is very robust and complicated, and it seems that most of the particularities of it are very subjective—they necessitate intimate familiarity with a specific system before any decent judgment can be made. Yet still, there are quite a few use cases for students, professors, and curriculum makers where cloud computing systems can be employed to streamline just about every deliverable process known to science. A cloud can solve a lot of issues but still be very costly, and nobody but a system admin can provide a very "good" answer as to how costly that could be in the short term, and regardless, it's very reasonable to assume from current data trends that most schools are likely to join the cloud trend, and that cloud migration is going to steadily increase moving forward to at least some degree.

Are cloud computing systems a "good idea" for a university? Hesitantly, yes, for a few different reasons. Firstly, a cloud computing system, while difficult to adequately price out, has been proven to be cost-effective monetarily in the long run due to the reduction in pricing involved with shifting maintenance costs onto third parties. Not only that, but they can also be cost-effective for processing because they handle the majority of their transactions on systems that a university doesn't have to host. Of course, these details may not necessarily be enormous concerns considering gains in cost may be marginal compared to the system being migrated *from,* so, secondly, it should be noted that cloud computing systems are beneficial to universities because of the benefits they bring to the ease-of-access department, if nothing else. As it has been shown, clouds that currently exist have already been applied to classrooms and proven to be effective in improving deliverable feedback and technological literacy for students, not to mention the ease of software distribution for professors and for even entire departments. These benefits do not exist in a vacuum, however, and thirdly it should be noted that trends in current data, as modeled by this very paper, indicate that a university's very status lends it to be likely to migrate to a cloud anyway considering how many already have, meaning that a migration away from a traditional server or even an outdated cloud is a great way for a university to stay with the times and give students experience with relevant, modern systems.

Are there drawbacks? Certainly, yes, the largest of which is the uncertainty with which a proper, long-term cost in terms of capital can be figured for a system. As discussed earlier, there are a great many ways to model the price of a cloud adoption that can be applied to a specific system should one know its details, but these can be difficult to make descriptive for specific architectures, and not all of them are even tested robustly enough to make them useful for newer systems. However, keep in mind that some cloud computing systems (such as Microsoft's technologies,) often include quite a lot of free software access and data storage, making it possible to negate some of the monetary costs that some universities may encounter. It is also worth remembering that some institutions may have no need for a dedicated cloud computing system, such as a school that specifies in a field with no real "quantitative methods" [5] or a school with very few students. For such cases, a cloud system could be very beneficial for record keeping or data storage, but may serve little other purpose and end up yielding plenty of wasted time for a full migration.

To be certain, though, the investigation has yielded a definitive answer by the "yes or no" schema, and that answer would be "yes." It is clear that cloud computing system can be beneficial to a university in many cases, and is not likely deleterious enough to system performance or department budget to make it a prospect worth fearing. It is obvious that many universities have already adopted cloud computing systems in one capacity or another, and as time moves on more of them will, and more technologies with more specific benefits will come along that will necessitate more migrations. To answer the quandary posed in the abstract, the average university *is* sufficiently large enough that it would be worthwhile to pay for these technologies, and many universities already have. Whether it is secure or not may be a question for a paper of a much different scope, but it is undeniable that cloud computing systems are the unavoidable near future for universities in America.

REFERENCES

1. B. Kim and G. Henke, "Easy-to-use cloud computing for Teaching Data Science," *Taylor & Francis Online,* 22-Mar-2021. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/10691898.2020.1860726?src=. [Accessed: 25-Jan-2023].
2. A. S. Theobold, S. A. Hancock, and S. Mannheimer, "Designing data science workshops for data-intensive environmental science research," *Taylor & Francis Online*, 22-Mar-2021. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/10691898.2020.1854636. [Accessed: 25-Jan-2023].
3. M. D. Beckman, M. Çetinkaya-Rundelb, N. J. Horton, C. W. Rundel, A. J. Sullivan, and M. Tackett, "Implementing version control with git and github as a learning objective in statistics and data science courses," *Taylor & Francis Online*, 22-Mar-2021. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/10691898.2020.1848485. [Accessed: 25-Jan-2023].
4. T. Antohl, "Model for Cloud Migration Cost," *IEEE Xplore*, 03-Oct-2019. [Online]. Available: https://ieeexplore.ieee.org/document/8854043. [Accessed: 24-Jan-2023].
5. Z. J. Kunicki, N. Z. Zambrotta, M. C. Tate, A. R. Surrusco, M. M. Risi, and L. L. Harlow, "Keep your stats in the cloud! evaluating the use of Google Sheets to teach quantitative methods," *Taylor & Francis Online*, 07-Oct-2019. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/10691898.2019.1665485. [Accessed: 25-Jan-2023].
6. Y. A. M. Qasem, R. Abdullah, Y. Y. Jusoh, R. Atan, and S. Asadi, "Cloud computing adoption in Higher Education Institutions: A systematic review," *IEEE Xplore*, 10-May-2019. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8712496/. [Accessed: 25-Jan-2023].
7. L. S. Vailshery, "Office 365 user numbers by country 2022," *Statista,* 23-Feb-2022. [Online]. Available: https://www.statista.com/statistics/983321/worldwide-office-365-user-numbers-by-country/. [Accessed: 25-Jan-2023].
8. C. Smith, "28 amazing amazon web services statistics and facts," *DMR*, 08-Jan-2023. [Online]. Available: https://expandedramblings.com/index.php/amazon-web-services-statistics-facts/. [Accessed: 25-Jan-2023].

9. “Instructure Usage Statistics,” *BuiltWith*. [Online]. Available: https://trends.builtwith.com/link/Instructure. [Accessed: 25-Jan-2023].
10. “Trends in Cloud Computing in Higher Education,” *eschoolmedia,* Jun-2016. [Online]. Available: https://eschoolmedia.com/wp-content/uploads/2016/06/vion0622.pdf. [Accessed: 27-Jan-2023].
11. C. Wu, A. N. Toosi, R. Buyya, and K. Ramamohanarao, “Hedonic Pricing of Cloud Computing Services,” *IEEE Xplore*, 23-Jul-2018. [Online]. Available: https://ieeexplore-ieee-org.proxy.ulib.uits.iu.edu/document/8418402. [Accessed: 12-Feb-2023].
12. I. Petri, J. Diaz-Montes, M. Zou, T. Beach, O. Rana, and M. Parashar, “Market Models for Federated Clouds,” *IEEE Xplore*, 23-Mar-2015. [Online]. Available: https://ieeexplore-ieee-org.proxy.ulib.uits.iu.edu/document/7065299. [Accessed: 12-Feb-2023].
13. “Canvas by Instructure: World's #1 teaching and learning software,” *Instructure*. [Online]. Available: https://www.instructure.com/canvas. [Accessed: 28-Feb-2023].
14. “Competitive intelligence for lead generation & qualification,” *HGData*. [Online]. Available: https://discovery.hgdata.com/. [Accessed: 09-Apr-2023].
15. “Data Home: College Scorecard,” *College Scorecard,* 14-Sep-2022. [Online]. Available: https://collegescorecard.ed.gov/data/. [Accessed: 09-Apr-2023].