Cloudifying the Curriculum with AWS

1st Michael Soltys
Computer Science
California State University Channel Islands
Camarillo, USA
michael.soltys@csuci.edu

***Abstract*—This is an Innovate Practice Full Paper. The Cloud has become a principal paradigm of computing in the last ten years, and Computer Science curricula must be updated to reflect that reality. This paper examines simple ways to accomplish curriculum cloudification using Amazon Web Services (AWS), for Computer Science and other disciplines such as Business, Communication and Mathematics.**

***Index Terms*—Computer Science, Internet-based instruction**

I. Introduction

Whether aware of it or not, most computer users have moved to the Cloud in the last fifteen years. They have done so by using Webmail, Google Docs, photo-sharing services or on-line gaming. Business has followed the trend a few years later, by replacing or supplementing in-house data centers with cloud services.

Cloud Computing is a new paradigm and also an inversion of an old one. When the telegraph was invented in the early 1800s, it was "smart at the edges" (the unit and its operator), while the connecting network was very simple (just a cable and repeaters [1]). Then came the telephone, with dumb terminals (rotary phones) at the edges, and a complex network inside with switchboards for circuit-switching. Then the paradigm was inverted once again with the Internet, where now we had smart terminals at the edges, and a relatively simple packet-switching network inside. The simplicity of the network allowed for the quick innovation that followed, and the result of that innovation was to invert the paradigm once again with Cloud Computing, where now computers at the edge are simply entry portals into a complex network, i.e, "the Cloud" [2].

At Universities, Computer Science (CS) and Information Technology (IT) faculty have been using the Cloud just like everyone else, but many have realized over the last decade that they can access compute power in the cloud without making large capital investments on campus, and can start using services, such as AWS virtual computers, with ease and speed. Since faculty work on research projects with students, senior students had to acquire cloud skills as well.

At the same time, job sites such as LinkedIn listed knowledge of the Cloud as the top job skill over the last five years [3], and currently 14% of *all* job listings require some understanding of the fundamentals of the cloud. As students and their families invest in costly education, they are keenly aware of this job market reality [4] (see blog entry [5]), and especially of AWS' growing dominance in this field [6]. Therefore, a demand for cloud instruction arose on campuses, and some faculty started adding cloud content to the curriculum. One of the earliest adopters of a cloud curriculum was the Santa Monica College [7].

This paper examines methods to cloudify the curriculum with AWS [8]. It starts from the assumption that the CS and IT curricula are well formed and mature, and that radical changes are not desirable. Instead, non-invasive and easy to implement ideas are proposed. Furthermore, the paper examines how cloud content can be presented to students in other disciplines, in particular students in Business, Communication and Mathematics. We concentrate on four-year Universities, while we are aware the two-year Community Colleges often led the charge in cloudification.

It should be mentioned explicitly that there are many cloud providers, and that universities and other institutions may find it advantageous to work with different ones; for example, Microsoft Azure won the JEDI (Joint Enterprise Defense Infrastructure) contract with the DoD (Department of Defense), although the award process was contested by AWS in court, and therefore the DoD may prefer the Microsoft Azure cloud when partnering with educational institutions. Another example is Machine Learning, where the leading technologies, MXNET, PYTORCH and TENSORFLOW (all three frameworks are open source), developed by Apache, Facebook and Google, respectively, also cross the boundaries between cloud providers. Does this mean that educational institutions need to teach the syntax of *all* leading cloud providers? Not necessarily, as the concepts are universal, and sometimes almost identical (e.g., both AWS and the Google Cloud have "Identity Access Management" services, with the same name, and very similar characteristics). This paper is written from the perspective of using AWS.

A few years ago we started Engineering on our campus at California State University Channel Islands, with Mechatronics Engineering. We found it very helpful to discuss with other Universities their experience in starting Mechatronics, and we found it especially helpful to consult the experience of the University of Utah which was recorded in [9]. The goal of this paper is to provide a similar template but for cloud adoption. The paper is meant to offer suggestions, rather than definitive solutions, and represents the opinions of the author. Each institution should adopt the cloud according to its particular circumstances.

II. CLOUDIFYING THE CURRICULUM

At the beginning of the discussion on curriculum cloudification we decided to examine the students we intended to serve, and we grouped them as follows:

1) Computer Science and Information Technology undergraduate majors.
2) Masters in Computer Science.
3) Business, Communication and Mathematics majors.
4) Working professionals.

As these students are served by different curricular pathways, this allowed us a divide-and-conquer approach to cloudification. And by "serve" we mean that we expose them to the Cloud to the degree they want, so that they can take advantage of the cloud-favorable job market.

A. CS and IT

The first group, CS and IT students, form our largest contingent of students, about five hundred currently, but growing quickly, as we have doubled our number of such majors over the last three years. About 80% of those students are in CS.

1) CS: Our CS curriculum is based on the ACM curriculum 2013 [10], where the four main pillars of AWS, *Compute*, *Storage*, *Databases* and *Networking*, are covered in depth. For example, Compute and Storage are covered in a sequence of three "systems" classes, COMP 162, 262 and 362, that cover everything from computer architecture and assembler to advanced topics in operating and file systems.

Databases are covered in our senior course COMP 420, and Networking in our senior course COMP 429. CS students also receive a solid grounding in programming capped with a junior Software Engineering class COMP 350.

Thus, our senior CS students are conceptually more than ready to study for the AWS Solutions Architect certification, and in fact any of the AWS certifications, except the professional one that requires some years of practical experience.

A light-footed approach to deliver cloud content to CS majors is to "sprinkle" it throughout our undergraduate classes as use cases and illustrations of concepts being taught. This approach **relies on the faculty discretion to decide on examples and platforms**,[1] and does not require time-consuming curricular changes and paying attention to accreditation issues with ABET[2] and WASC[3]. Core concepts are presented in class as promised in the syllabus, but the illustration of those concepts with AWS tools and use cases falls under the discretion of the instructor.

Some instructors may choose to use AWS, some may not. However, for all students interested in absorbing more cloud material we offer two ways to do so: one, to use AWS in their capstone (a full year class in senior year), two, to take a special topics COMP 490 class that follows the AWS Solutions Architect certification. Both ways may be chosen.

Of course, as cloud content becomes embedded in the curriculum in the upcoming years, reflecting the new paradigm in the industry, some instructors may choose to formalize it in the syllabi, possibly requiring curriculum committee approvals; however, this does not preclude teaching the material immediately under our "faculty discretion" model explained above.

2) IT: In our school the IT program was brought about in order to accommodate transfer students from local colleges. The program was built using a grant, and at the time of its founding it was designed to emphasize web development. The principal difference between CS and IT is that IT students take less mathematics classes, less programming, and have more electives (partly to assist in the transfer that would allow their community college classes to count toward graduation).

In recent years we have been looking for a new, more current, emphasis for our IT program, and we considered business or cybersecurity. But we started discussing the possibility to make it Cloud centric by adding AWS to the curriculum more deliberately. This would require better coordination with Community Colleges. However, at this moment, the approach to offering AWS to our IT students is similar to CS: the choice to take an AWS special topics class in the senior year.

B. Masters in CS

In the Masters program in Computer Science, we introduced the cloud into three core classes: Networking, Security and Software Engineering.

1) Networking: For the last decade we offered a graduate course in Networking with the title "Cloud Computing" (COMP 529) [12]. This course is a graduate version of our undergraduate Networking class (COMP 429). Recently we introduced the material from the AWS Solutions Architect (SA) certification into its syllabus. The symbiosis of Networking and AWS is a very natural way to deliver this class, and students get an offering that not only covers the entire SA certification, but goes far beyond it. For example, students learn about routing at the IP level, and flow and congestion algorithms at the TCP level, as well as most of the content in [13].

2) Security: Cybersecurity is one of the areas of emphasis in our department, and we have been teaching a graduate version of Cybersecurity (COMP 524) for several years [14]. (The undergraduate version of that course is COMP 424.) Since the author received an AWS Specialty Security (SS) certification in December 2019, we have been updating the class material to include the entire certification curriculum as use cases and examples. The author wrote a manuscript containing the notes for the class, with advanced material such as Cryptography, Malware, Distributed Denial of Service attacks, as well as an overview of most of the tools in Kali Linux.

[1] See [11] for a discussion of the meaning of "Academic Freedom." Keep in mind that an instructor is responsible for covering the content in the syllabus of a course, as the syllabus is a contract between the university and the student. However, point 11 of [11] says that "Academic freedom gives faculty members substantial latitude in deciding how to teach the courses for which they are responsible." In our case, instructors can illustrate the concepts with examples, use cases and technologies of their choosing, especially since syllabi tend to be technology agnostic.

[2] https://www.abet.org/

[3] https://www.acswasc.org/

Adding the material in the AWS SS certification allows us to update the course to a more current offering as it now includes security in the Cloud. This includes details of Identity Access Management (IAM), S3, Security Policies, Logging and Monitoring, Key Management System (KMS) and use cases of CloudTrail, CloudWatch, Inspector, Cloud Formation, Cloud Config, and other AWS tools that allow a more hands-on approach.

The class has really improved as the result of including the AWS SS material. For example, it is one thing to talk about the importance of data encryption at rest for PCI compliance; it is quite another to demonstrate to the students the automated enforcement of such compliance with Amazon Macie or Config.

3) Software Engineering: We invested quite heavily in our Software Engineering offering, both at the undergraduate and graduate levels. A significant portion of our students become employed as programmers, and we want them to be team leaders and to think as engineers in their future roles in the industry. Programming and Software Engineering skills are in great demand by employers.

Following the arguments in [15], we base our design of Software Engineering on the following insights: (i) in the first phase of cloudification, the cloud revolutionized system administration, and now in the second phase it is simplifying and changing cloud programming. (ii) A significant portion of development is now done using de-coupled micro-services; thus, using the AWS cloud we teach an approach to application development that is serverless ("Function-as-a-Service" — FaaS).

We base our approach on the Agile methodology, and our goal is to combine the wisdom contained in the literature, e.g., [16]–[20], with the tools used in the industry (such as Atlassian's BitBucket and Jira, or GitHub), and with the AWS Cloud as a background that allows the students to gain hands-on experience.

C. Business, Communication and Mathematics

It is easy to include Mathematics majors in our cloudification initiative, as CS and Math are closely integrated as departments, and our majors are not far from being double-majors. Hence what we wrote about CS students in Section II-A1 applies directly to Math students. Mathematics also delivers emphases in Data Analytics and Imaging, both of which can take advantage of AWS tools such as Amazon Athena, Redshift, Rekognition, etc. We plan to introduce these tools into Mathematics.

As was mentioned in the Introduction, LinkedIn has listed the Cloud as a top job skill over the last five years (in 2020 the top spot may be replaced by Blockchain [21], but even so the Cloud will be a close second). However, the top *soft skills* include persuasion, collaboration, adaptability and emotional intelligence, associated with Business and Communication students. As mentioned in the Introduction, 14% of jobs will require some cloud knowledge, and so Business and Communication students can expect to work in environments where the Cloud will be central to a business mission.

We have designed a new course, *Online Communication and Society*, COMP 347, cross-listed with Communication, where we have embedded the Cloud with Business and Communication students in mind [22]. In this class, the students learn the basics of AWS, launch an EC2 instance, and install a LAMP stack on it in order to install WordPress. Once WordPress is running, they integrate it with social media (e.g., LinkedIn and Twitter), and publish content as posts and pages.

In order to make the exercise more practical, we propose to the students as framework the following use case: *you have been hired by a non-profit with few resources but big ambitions, and you need to promote your mission with a minimal budget*. The content was developed by the author, and the technical material was based on AWS excellent whitepapers, in particular [23], [24].

It is important to remember that for most companies, the decision to move to the cloud is principally a business decision, not an IT decision. Both the basic AWS Cloud Foundations certification, and the advanced AWS Solutions Architect certification emphasize the business aspect of cloud solutions. This is a natural arena for business students, and as the CEO of AWS mentioned at 2019 re:Invent in Las Vegas, only 3% of world IT is currently in the cloud (while it comprises over 60% of all IT spending!), and so as more IT moves to the cloud, this will be an area of activity for business oriented students.

One more thing ought to be mentioned: we have joined the *AWS Activate for Startups* program, where we can nominate startups for low cost AWS credits. This works well with our campus entrepreneurship initiative.

D. Working professionals

We are offering two certification classes open to non-matriculated students (i.e., anyone who wants to take them): AWS Cloud Foundations and AWS Solutions Architect, both of them through AWS Academy [25]. These classes are aimed at working professionals, they are eight weeks long, and delivered mostly online. We will seek approval for a Cloud Certificate through our campus curriculum committee. The mechanism for delivering this certificate program is our Extended University[4], which does not receive State funding, and is therefore more flexible in how it can mount programs.

The Cloud certificate is a service to our community, especially the Navy (there are two large Navy bases in Ventura County), as well as the industry comprised of the Department of Defense contractors as well as companies in the "101 Technology Corridor." Furthermore, as [26] writes in his paper *The great enrollment crash*, most universities are expecting up to a 15% drop in enrollment due to demographics over the next decade, and serving the working population will become only more important to our viability as an institution.

[4]Extended University is often called the "School of Continued Education," but diplomas from EU are simply diplomas from our school. It is an internal administrative entity, but part of the university as such.

III. Example with Machine Learning

In this section we give a more detailed example of how we incorporate the cloud in our offering: we show how we approach teaching Machine Learning (ML) using the AWS cloud. The key point is that for a small school like ours, partnering with AWS allows us to teach ML in a hands-on and cost-effective manner.

The AWS ML requires the mastery of AWS IT infrastructure, eg., Kinesis; Statistics, eg., Principal Component Analysis (PCA); an expert level familiarity with AWS SageMaker, a one-stop shop for ML in the AWS console; and modeling with a large variety of algorithms, e.g., XGBoost, K-NN, Linear Learner, etc. In short it requires background in Machine Learning, especially in model tuning, in Statistics, and in the AWS cloud.

Marvin Minsky and Seymour Papert's [27], written in 1969 with a second printing in 1972, is one of the first books in ML. The subtitle of the book is *An Introduction to Computational Geometry*, and the emphasis of this visionary book, which laid the foundations of ML, is on what we would call today object recognition. Indeed, the book's approach is through the problem of computer vision. Many of the mathematical ideas proposed in the book lay dormant for years, as the hardware needed to run the required computation did not exist at first, and later was the domain of a few scientists with access to mainframes. But today, thanks to the economies of scale of computing allowed by the cloud, anyone can set up a ML training job for a few hundred dollars. Of course, this led to an explosion of the field, and its great applicability in medicine, finance, weather predictions, recommender systems, etc. However, ML is still expensive, and our school was able to take advantage of an AWS ML Pilot program that made the technology available to our students at no cost.

For CS majors, ML is a lot more than the mathematics of perceptrons, which has been the traditional way of teaching the subject without access to a suitable computing environment. In particular, there are four areas of ML of concern to Computer Science:

1) Data Engineering
2) Exploratory Data Analysis
3) Modeling
4) ML Implementation and Operations

which also correspond to the four areas in the AWS ML certification (a difficult exam). We emphasize once again that a CS approach to ML requires all four areas above; otherwise, one ends up with a course in the theory of ML rather than the full range of this rich field. An excellent cost-free textbook that balances theory and practice is [28], and it complements very well the material provided by AWS in the form of whitepapers and guides.

Data Engineering covers how to create data repositories, eg., a Data Lake, how to ingest data into the repository, and then how to transform the raw data in the repository into data that can be analyzed. This last step is known as a data cleaning operation.

An example of a typical solution is an S3 bucket that hosts the Data Lake, a massive dump collecting data from various sources. This data is then worked on by an Extract-Transform-Load (ETL) application hosted on an Elastic Map Reduce (EMR) cluster. The type of work done here consists in taking the data from a heterogeneous set of sources (JSON files, text files, Relational Data Base dumps, images, etc.), and making the data uniform, conforming to a set of conventions described by a table of items (rows) with attributes (columns). The ETL application is typically something like Apache Spark or Apache Hadoop. The data is then written back into the S3 bucket hosting the Data Lake (or possibly a new S3 bucket). The data is now ready, and commonly living in a Pandas table, for consumption by the second stage of the pipeline: Exploratory Data Analysis.

In the Exploratory Data Analysis, data is sanitized. This can mean, for example, taking a variety of date formats (eg., January 23, 2021, 01/23/2021, 23-1-21, etc.) and making them uniform; or, dealing with missing or incomplete data.

ML takes numeric data only, and so all columns have to contain numbers. This may mean taking ordinal data (such as large, medium, small) or nominal data (such as colors), and representing it with numerical data. A subtle point here is that large, medium, small may be translated into 3,2,1, as the numbers represent the order, but the same should not be done with colors where there is no natural ordering to them. In the case of colors, we may want to use one-hot encoding: replace the column with color names with a set of columns with headings for the different colors, as in this example:

Color	Red	Blue	Green
Red	1	0	0
Blue	0	1	0
Green	0	0	1

There are two techniques for dealing with missing data: (i) Remove rows or columns containing missing data. (ii) Impute missing values, that is, interpolate the missing values by replacing them with the mean of the column, or simply zero, or a little bit more advanced, using regression, very commonly linear regression, which assumes that the data can be estimated well from a linear combination of the existing entries.

Once the data is sanitized and resides in a table it is possible to perform feature engineering on it. Note here that there are two related terms: attributes and features. Attributes come with the raw data; they are given. Features, on the other hand, are those attributes we select to run the training process. Part of feature engineering is to select appropriate attributes for features.

For example, in detecting fraudulent credit card transactions, it may not be necessary to include the transaction id, which may be generated randomly and carry no information needed to train a model. Thus we drop the transaction id attribute, i.e., it does not become a feature. Another example may be when predicting school enrollments (which students end up coming in the fall based on their applications), it may not be necessary to have both SAT and ACT scores; the two scores may be highly correlated, in which case we only need one of them to train the model. The correlation of two attributes may be found visually with a correlation matrix or a scatter matrix.

The process of removing attributes and/or combining them is called dimensionality reduction, and there are two techniques for it: (i) Principal Component Analysis (PCA), and (ii) t-Distributed Stochastic Neighbor Embedding (t-SNE). It may be confusing that PCA and t-SNE are ML algorithms, in fact unsupervised ML algorithms, which are also used to prepare data for ML training.

Modeling: As with any advanced field in the sciences, it is good to be familiar with the nomenclature, in particular the difference — in the context of ML —- between algorithm, model and framework:

In the context of ML, algorithms take as input your data and output a model. For example, in the case of the Linear Regression algorithm, the input is a set of observations, and the output is a linear function with a bias.

A Model is the interpretation of the data obtained for the sake of predictions, computed with an algorithm. The quality of a model depends on the selection of an appropriate algorithm, the fine tuning process, and the testing. Framework: it is a set of software tools, libraries and interfaces used to compute models with algorithms. For example, the textbook Dive into Deep Learning, proposes three frameworks: MXNET, PYTORCH and TENSORFLOW, all three frameworks are open source, developed by Apache, Facebook and Google, respectively. SageMaker supports all three frameworks, as well as others.

We should also add tensors to the list useful definitions. There seems to be confusion regarding what is a tensor. The confusion may arise from the generality of the concept; many mathematicians have learned the concept in a book such as Spivak's Analysis on Manifolds; physicists use tensors in mechanics. For the sake of ML, the most useful way to conceptualize tensors is as a generalization of the sequence: scalar (0-dim tensor), vector (1-dim tensor), matrix (2-dim tensor), and now a "cube matrix," indexed by (i, j, k) is a 3-dim tensor, etc. Indeed they are called ndarray in MXNet. Tensors facilitate the designation of linear transformations in n-dimensional vector spaces.

The primary interface to SageMaker is the Jupyter Notebook, a development environment popular with Data Analysts. A Jupyter notebook has a kernel which is the computational engine on which the notebook runs, and both Python and R are naively supported on SageMaker notebooks. When opening a new Jupyter Notebook in SageMaker, the user can select a kernel which supports a given framework.

The first step in the area of Modeling is to cover the different SageMaker algorithms. The Table: Mapping use cases to built-in algorithms in the linked document is especially useful, as it lists the algorithms according to use cases. We summarize them here:

Supervised Learning:

- Binary or Multi-class classification; eg., a spam filter
- Regression; eg., estimating home values
- Time Series Forecasting; eg., predict sales; the algorithms here are DeepAR for Time Series, and for the first two, classification and regression, they are: Factorization Machines, K-NN, LL, XGBoost

Unsupervised Learning:

- Feature Engineering (PCA); eg., combine several features into one (component)
- Anomaly Detection (RCF); eg., find outliers that make model training more difficult, since the "regular" points without outliers lends themselves to a simpler model
- Embedding high to low dimension (Object2Vec)
- Clustering or Grouping (K-Means); for discrete groupings within data
- Topic Modeling (organize docs into topics not known in advanced: LDA, NTM)

Textual Classification:

- Text classification into pre-defined categories (BlazingText)
- Translation, summary or speach2text (Seq2Seq)

Image Processing:

- Image and multi-label classification
- Object detection and classification
- Computer Vision, as in self-driving cars

Hyperparameters are values given to a particular ML algorithm that control its runtime: size of steps, constants, batch sizes, etc. Choosing the right hyperparameters is an art form, and it is acquired by experience. Note that just as we used ML for dimensionality reduction, we can also use ML to tune hyperparameters – this is done with Bayesian search and explained in the link just given. In order to help with the understanding of hyperparameters, I recommend reading Linear Regression Implementation from Scratch, where basic hyperparameters such as learning rate, minibatch size, epoch and gradient ascent rate are explained with great examples.

A framework has been selected, and algorithm chosen, a model constructed; how do we know the quality of the model? We need to be able to test against some data that was not used in the training, but for which we know the targets. To that end, it is customary to train the model on 80% of the data, and reserve 10% for validation, used while building the model, and 10% for testing, done at the end. Let's concentrate on the supervise binary classification case.

The first step of testing is to build a confusion matrix, which counts how many True Positives (TP), False Negatives (FP), False Positives (FP) and True Negatives (TN) there are. For example, TP are those items which were predicted as positive by the model, and were actually positive; FP are those items which were predicted positive erroneously by the model, since they are in fact negative. There are several metrics associated with a confusion matrix. In the diagram to the left we show Precision, which is TP/(TP+FP), and Recall, which is TP/(TP+FN). Imagine that our model is classifying MRI images according to whether cancer is present or not. It is desirable to identify all images with cancer, even at the cost of having some false positives (the argument being that a dramatic scare is preferable to a tragic neglect of treating a cancer). Thus, in this case high recall can be striven for at the cost of a moderate precision. Note that both recall and precision have a value in [0,1], and a recall close to 1 implies a FN close to zero.

Implementation and Operations: A ML model exists, and now we want to deploy it. In this domain Containers make an appearance, and some familiarity with the concept is necessary (containers are covered in detail in the AWS Developing certification). Another fundamental deployment concept is that of an endpoint. An endpoint is where we attach the model once ready to make inferences; it is a fully managed service that allows real-time inferences via a REST API.

Deployment of a model: The components of a typical deployment, going right to left in the left figure, is an API managed by AWS API Gateway, that connects to a model endpoint, created with a single line of code in, say, a Python SDK, which then connects to a load balancer which distributes the queries to the model hosted in containers on EC2 instances, possibly spread in several Availability Zones, and part of an auto-scaling group. Note that when a SageMaker model is being trained it is housed in a container, and then the same container, but now with parameters set post-training, is used in the deployment. Also note that endpoints are flexible, in the sense that it takes relatively little effort to have more than one model behind an endpoint (using a shared serving container), with A/B testing supported. The multi-model deployment on a single endpoint, rather then several endpoints, is a good answer to a question that emphasizes low deployment cost, as it reduces deployment overhead since SageMaker manages loading models in memory and scaling them based on the traffic patterns.

The above endpoint would work on sporadic queries, but what if the batch or streaming predictions are required? For batch predictions, raw data may be put in an S3 bucket, transformed by an ETL process (EMR + Apache Spark, or Glue), and using a batch transform ingested by the model (note that one of the advantages of batch transform is that you can feed batch data to a model without deploying a persistent endpoint). For streaming predictions, data may be ingested by Kinesis, which connects with SageMaker.; see this blog post. Also see this blog post on Kinesis ingestion of video streams.

IV. Ancillary Efforts

In pursing a partnership with a cloud provider, the largest overhead is the time required to train the instructors. Most faculty have graduated from school before the cloud revolution, and while most have the required background (distributed algorithms, networking and "IT as utility"), there is a substantial effort to learn a particular cloud technology. There is also the need to communicate the benefits of this effort to the administration. We summarize this background work in this section.

Instructor training and certification: there has to be a critical mass of instructors, both tenure-track faculty and lecturers, who are willing to learn and deliver the AWS content. We currently have two AWS Academy accredited instructors, and three more in the pipeline. The author is the principal point of contact for AWS Academy on campus, and can nominate interested educators.

Communicate to the faculty the benefits of Cloudifying: tenure-track faculty have invested many years into their careers: a PhD, then post-docs, then working toward tenure, this can be 15 years or more. At the same time, in an era of great specialization, they work to become respected members of a research community. Their time to learn the foundations of other areas is very limited. One way to motivate them is to present the advantages of AWS in their research; this is one of the aims of the AWS Ambassador program.

Curriculum Changes vs Faculty Discretion: As already discussed in the paper in Section II-A1, it may be more effective at first to invoke faculty discretion in introducing cloud content rather than an effort to codify a new curriculum. Curricular decisions are always lengthy, take up several teaching cycles, and the department may end up missing the boat of cloudification. We argue that it is best to cloudify *now* with faculty discretion, while pursuing curricular changes as they become necessary. This provides a mechanism to synchronize the different speeds of innovation in industry versus the academia.

Advisory Board: a departmental advisory board can be a great ally in bringing about the cloudification of the curriculum. For one, the members of the board are probably contemplating a possible move to the cloud, and understand the need to train the workforce. In our case we are lucky to have a large and supportive advisory board, comprised of about 20 local companies, and we kept them abreast of the AWS initiative from the beginning. [29]

Support of the administration: faculty have ten great ideas every day, but they all require precious and limited campus resources. It is imperative to have the support of the administration while pursuing this initiative, which, as in the case of faculty, requires communicating the benefits and low costs of the effort. In our case we were very fortunate to have such support. It helped to receive a $35K credit grant from AWS.

V. Conclusion

Computer Science is a field with a fast paced proposal of new paradigms, and curricula must, on the one hand, offer the fundamentals of a now well establish field, but on the other hand be nimble enough to accommodate the fast rate of innovation. One way to accomplish this is to offer a core of fundamentals in each area, illustrated with use cases from the latest trends. We propose that this is a practical and effective approach to introduce Cloud Computing into the curriculum.

In order to cloudify the curriculum, a university needs to enter into a partnership with a cloud provider. While there are many cloud providers, this paper is written from the perspective of a partnership with AWS, which has an excellent interface with colleges through its Academy and Educate initiatives. The benefit to the faculty is a rich curated curriculum that can be easily incorporated into the standard CS and IT offering. The benefit to the students is that they learn the new cloud paradigm in a hands-on manner, being able to test all concepts immediately and at a very low cost.

VI. Acknowledgments

We are grateful to the *AWS Academy* and *AWS Educate* teams for guiding us through the process, and for all high quality educational materials they offer for free to participating institutions. We are also grateful to Michael Berman for comments on an early draft.

References

[1] C. Petzold, *The Hidden Language of Computer Hardware and Software*. Microsoft Press, 2000.

[2] P. Castro, V. Ishakian, V. Muthusamy, and A. Slominski, "The rise of serverless computing," *Communications of the ACM*, vol. 62, no. 12, pp. 44–54, Nov 2019. [Online]. Available: http://dx.doi.org/10.1145/3368454

[3] P. Petrone, "The skills companies need most in 2019 — and how to learn them," *Linkedin The learning blog*, January 2019. [Online]. Available: https://learning.linkedin.com/blog/top-skills/the-skills-companies-need-most-in-2019--and-how-to-learn-them

[4] L. Columbus, "15 top paying it certifications in 2019," *Forbes*, 2019. [Online]. Available: https://www.forbes.com/sites/louiscolumbus/2019/02/11/15-top-paying-it-certifications-in-2019/#5821321d3e7c

[5] M. Soltys, "Winter class at CI in Cloud Computing," online, October 2019. [Online]. Available: http://prof.msoltys.com/?p=5123

[6] ——, "Tech jobs: Python programming language and AWS skills demand has exploded," online, November 2019. [Online]. Available: http://prof.msoltys.com/?p=5224

[7] "SMC launches cloud computing certificate in high-demand IT field," May 2018. [Online]. Available: http://www.smc.edu/NewsRoom/Pages/Cloud-Computing-Certificate.aspx

[8] Amazon Web Services, "Overview of amazon web services," AWS, Tech. Rep., October 2019. [Online]. Available: https://d1.awsstatic.com/whitepapers/aws-overview.pdf

[9] S. Meek, S. Field, and S. Devasia, "Mechatronics education in the department of mechanical engineering at the university of utah," *Mechatronics*, vol. 13, pp. 1–11, 2003.

[10] "Computer science curricula 2013: Curriculum guidelines for undergraduate degree programs in computer science," 2013. [Online]. Available: https://www.acm.org/binaries/content/assets/education/cs2013_web_final.pdf

[11] C. Nelson, "Defining academic freedom," December 2010. [Online]. Available: https://www.insidehighered.com/views/2010/12/21/defining-academic-freedom

[12] M. Soltys, "COMP 529 – Cloud Computing – spring 2019," online, May 2019. [Online]. Available: http://prof.msoltys.com/?page_id=4255

[13] J. Kurose and K. Ross, *Computer Networking: A Top-Down Approach*, 7th ed. Pearson, 2017.

[14] M. Soltys, "COMP 524 – Cybersecurity – Fall 2018," online, December 2018. [Online]. Available: http://prof.msoltys.com/?page_id=3505

[15] J. Schleier-Smith, V. Sreekanti, A. Khandelwal, J. Carreira, N. J. Yadwadkar, R. A. Popa, J. E. Gonzalez, I. Stoica, and D. A. Patterson, "What serverless computing is and should become," *Communications of the ACM*, vol. 64, no. 5, pp. 76–84, may 2021. [Online]. Available: https://doi.org/10.1145%2F3406011

[16] *Don't make me think*. New Riders, 2014.

[17] *The Mythical Man-Month*. Addison-Wesley, 1995.

[18] J. Ousterhout, *A philosophy of software design*. Yaknyam Press, 2018.

[19] R. C. Martin, *Clean Code*. Addison-Wesley, 2009.

[20] N. Forsgren, J. Humble, and G. Kim, *Accelerate: Building and Scaling High Performing Technology Organizations*, ser. The Science of Lean Software Devops. IT Revolution, 2018.

[21] D. L. Pate, "The skills companies need most in 2020 — and how to learn them," *Linkedin The learning blog*, January 2020.

[22] M. Soltys, "COMP 347 – online communication and society – summer 2019," online, May 2019. [Online]. Available: http://prof.msoltys.com/?page_id=4527

[23] "Tutorial: Install a LAMP Web Server on Amazon Linux 2." [Online]. Available: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-lamp-amazon-linux-2.html

[24] "Tutorial: Hosting a WordPress Blog with Amazon Linux." [Online]. Available: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/hosting-wordpress.html

[25] M. Soltys, "AWS training at CI in the spring 2020," online, November 2019. [Online]. Available: http://prof.msoltys.com/?p=5203

[26] B. Conley, "The great enrollment crash: students aren't showing up. and it is only going to get worse." *The Chronicle of Higher Education*, September 2019.

[27] M. Minsky and S. A. Papert, *Perceptrons*. MIT Press, 1969.

[28] M. L. A. J. S. Aston Zhang, Zachary C. Lipton, "Dive into deep learning," Tech. Rep., 2020. [Online]. Available: https://d2l.ai/

[29] M. Soltys, "Advisory Board," online, December 2019. [Online]. Available: https://prof.msoltys.com/?tag=advisory-board

[30] R. Lämmel, "Google's MapReduce programming model — revisited," *Science of Computer Programming*, vol. 70, no. 1, pp. 1–30, jan 2008. [Online]. Available: https://doi.org/10.1016%2Fj.scico.2007.07.001

[31] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, apr 2010. [Online]. Available: https://doi.org/10.1145%2F1721654.1721672

[32] D. Foster, L. White, J. Adams, C. Erdil, H. Hyman, S. Kurkovsky, M. Sakr, and L. Stott, "Cloud computing: developing contemporary computer science curriculum for a cloud-first future," in *ITiCSE*, 2018.

[33] Y. Demchenko, A. Belloum, D. Bernstein, and C. D. Laat, "Experience of profiling curricula on cloud computing technologies and engineering for different target groups," in *2014 IEEE 6th International Conference on Cloud Computing Technology and Science*. IEEE, dec 2014. [Online]. Available: https://doi.org/10.1109%2Fcloudcom.2014.160

[34] "How AWS came to be," *TechCrunch*, 2016.

[35] L. Chen, Y. Liu, M. Gallagher, B. Pailthorpe, S. Sadiq, H. T. Shen, and X. Li, "Introducing cloud computing topics in curricula," *Journal of Information Systems Education*, vol. 23, no. 3, Fall 2012.

[36] Y. Demchenko, D. Bernstein, A. Belloum, A. Oprescu, T. W. Wlodarczyk, and C. de Laat, "New instructional models for building effective curricula on cloud computing technologies and engineering," in *2013 IEEE 5th International Conference on Cloud Computing Technology and Science*. IEEE, dec 2013. [Online]. Available: https://doi.org/10.1109%2Fcloudcom.2013.160

[37] *Panel: Addressing the shortage of big data skills with inter-disciplinary big data curriculum*, 2019.

[38] *Panel: incorporating cloud computing competneces into computing curriculum: challenges and prespects*, 2021.

[39] *Panel: Integrating Requirements Engineering Education into Core Engineering Disciplines*, 2021.

[40] J. C. Nwokeji, S. T. Frezza, J. D. T. S. Holmes, and V. Uskov, "Panel: Software requirements engineering education in a changing world," in *2017 IEEE Frontiers in Education Conference (FIE)*. IEEE, oct 2017. [Online]. Available: https://doi.org/10.1109%2Ffie.2017.8190562

[41] I. Sommerville, "Teaching cloud computing: a software engineering perspective," School of Computer Science, St. Andrews University, Scotland, Tech. Rep.

[42] A. Ghoshal, E. C. Larson, R. Subramanyam, and M. J. Show, "The impact of business analytics strategy on social, mobile, and cloud computing adoption," *Thirty Fifth International Conference on Information Systems*, 2014.

[43] "The NIST definition of cloud computing," National Institute of Standards and Technology, Tech. Rep., 2011.

[44] Amazon Web Services, "Security pillar: AWS well-architected framework," Amazon, Tech. Rep., July 2018.

[45] J. Gabrielson, "Avoiding fallback in distributed systems," AWS, Tech. Rep., 2019.

[46] M. Brinkley and J. Chhabra, "Caching challenges and strategies," AWS, Tech. Rep., 2019.

[47] M. Brooker, "Leader election in distributed systems," AWS, Tech. Rep., 2019.

[48] C. MacCárthaigh, "Workload isolation using shuffle-sharding," AWS, Tech. Rep., 2019.

[49] R. Palmer and J. Kim, "Reaching for the cloud," *Ovum*, 2019.