Bureau of Labor Statistics

Time Dummy Hedonics for PPI Cloud Computing

Claire O'Bryan and Steven Sawyer

Introduction

Cloud computing is a key information technology service with large and growing revenue. Worldwide end-user spending on public cloud services is projected to grow 23.1% in 2021 to $332.3 billion from $270 billion in 2020, and it is forecast to increase to $397.5 billion in 2022.[1] Furthermore, cloud services are projected to make up 14.2% of total IT spending worldwide by 2024 up from 9.1% of spending in 2020.[2] Like many "high-tech" goods and services, cloud computing sees strong revenue growth while also undergoing rapid technological improvement. This improvement, also known as quality change, needs to be accounted for when measuring the rate of price change over time.

The Bureau of Labor Statistics' (BLS) Producer Price Index (PPI) tracks inflation by measuring only "pure" price change, which is price change that excludes changes in quality. When a product or service does change, there are methods, known as quality adjustment (QA), which can account for this change.

One of the QA methods is hedonic quality adjustment. This method uses a hedonic regression model to estimate and control for changes in quality in order to give a measure of pure price change.

For this article, BLS uses data from the top three cloud providers to develop a hedonic model for cloud computing services. We will cover: (1) cloud computing industry, index, and technological background, (2) data and methodology for the hedonic models, (3) construction of the model, and (4) the results of the model with comparisons to existing methodology.

Industry and Index Background

The PPI for Data processing, hosting, and related services surveys establishments that provide infrastructure or support for hosting or data processing purposes. The establishments are often third-party service providers for other businesses and governments who outsource their business processes, and/or data and computing services. These outsourced services are provided by equipment owned, operated, and held by the establishments within the data processing and cloud industry. This industry is highly concentrated and the very large providers have the greatest influence over the market prices for cloud services.

---

[1] "Gartner Forecasts Worldwide Public Cloud End-User Spending to Grow 23% in 2021", Gartner, April 2021, https://www.gartner.com/en/newsroom/press-releases/2021-04-21-gartner-forecasts-worldwide-public-cloud-end-user-spending-to-grow-23-percent-in-2021

[2] "Gartner Forecasts Worldwide Public Cloud End-User Spending to Grow 18% in 2021", Gartner, November 2020, https://www.gartner.com/en/newsroom/press-releases/2020-11-17-gartner-forecasts-worldwide-public-cloud-end-user-spending-to-grow-18-percent-in-2021

Amazon Web Services (AWS) leads the cloud computing industry with ownership over nearly half of the Infrastructure as a Service (IaaS) market (47.8%) in 2018. The other leaders in the industry in 2018 were Microsoft (15.5%), Alibaba (7.7%), Google (4%), and IBM (1.8%).[3] As leaders in the industry, their pricing structures are similar. AWS, Microsoft Azure, and Google all have customizable on-demand packaging as well as pre-structured packages that are comparable and classified or broken down by the same characteristics. For these reasons we have chosen to build our data sample using these three service providers.

Traditionally, business processes and data and computing services have been completed on-site by the businesses using them. For instance, data management and payroll used servers and computers that were co-located in the same building where a business conducted its primary activities. With the emergence of the data processing service industry, businesses could rent space and power for servers off-site.

There are several major benefits to outsourcing these services. First, businesses no longer have to be concerned with purchasing and maintaining complicated and expensive information technology (IT) equipment. Another benefit is that the services provided are often sold in an on-demand, customizable fashion. As companies' computing needs change, they can vary the capacity of data processing services they purchase, which can save costs. If the companies' were to host data processing services in-house, they would need to have computing resources available to handle peak demand that would otherwise sit idle most of the time. These services are measured and charged based on the time, storage, memory, etc. used by the client company and are typically outlined by a contract between the client and the providing establishment.

Within the PPI's index structure, Data processing, hosting, and related services is an aggregate index that contains lower level indexes. These lower level indexes are based on the types of services the items provide: Business process management services; Data management, information transformation, and related services; Hosting, application service provision (ASP); and other IT infrastructure provisioning services. Cloud computing is the provisioning of virtual computer infrastructure which is classified in the Hosting, ASP, and other IT infrastructure provisioning services index.

---

[3] Jeb Su, "Amazon Owns Nearly Half of the Public Cloud Infrastructure Market Worth Over $32 Billion: Report", *Forbes*, August 2019, https://www.forbes.com/sites/jeanbaptiste/2019/08/02/amazon-owns-nearly-half-of-the-public-cloud-infrastructure-market-worth-over-32-billion-report/#7f7c713d29e0

Cloud computing can be classified into three areas: software as a service, platform as a service, and infrastructure as a service. Software as a service (SaaS) is the most fully featured and easily accessible cloud computing package. SaaS is access to software online hosted by the service provider. With SaaS, there is no need for a customer to install, manage, or purchase hardware. The customer simply connects to the cloud provider and uses the software. Platform as a service (PaaS) has a much more broad structure compared to the polished SaaS packages. As its title suggests, PaaS provides a platform on which developers can build and attach their own applications. These platforms are typically made up of an operating system (OS), a programming language and an environment for it, a database, and a web server. Finally, the service we will focus on throughout this article, infrastructure as a service (IaaS), is the most basic type of cloud computing. Each package is essentially a virtual version of a blank computer: microprocessor, memory, storage, etc. Furthermore, these packages often include access to a basic OS, such as a limited version of Linux, or the option to purchase access to a preferred OS, such as Windows.

BLS chose to focus on IaaS packages for the PPI's QA model because, as the broadest service offering, IaaS is typically used as a base on which other services are built. For instance, SaaS and PaaS both use the computer resources that are offered through IaaS. By developing a model for IaaS, we are also able to describe many of the factors that affect the price for SaaS and PaaS.

BLS determines the current pricing method for IaaS in the PPI by both the service characteristics and transaction terms for each item. For IaaS, the main types of price are fee-based transaction prices (average rates, standard rates, or prepaid rates) or estimated flat fees. Flat fees are more commonly seen in contracts with large firms. These contracts are negotiated based on an average or expected sum of cloud usage per month. However, in actuality, cloud usage is so variable that the real value of these contracts is almost never the same month-to-month. Fee-based transaction prices have become much more common in the cloud computing industry because the industry has shifted toward on-demand services in order to cater to small businesses, individual consumers, and large companies simultaneously. To control for variability of usage, we focus on per-hour pricing in IaaS packages as the dependent variable of our model.

The service characteristics are the variable features, and the combinations of these features determine the price, either as a contract or a sum-of-its parts fee-based transaction. For IaaS packages, these service characteristics include, but are not limited to:

- Application support/customer support dedicated to the specific package
- Shared vs. dedicated/managed environment
- Microprocessor
- Operating system
- Memory
- Data storage
- Number of users
- IP address type (dynamic vs. static, number of addresses)
- Computer time used in a given period (rented, leased, shared)
- Training
- Management
- Region

The combination of the above characteristics that a customer may choose to purchase is highly customizable to fit the needs of any customer from independent users, small businesses, to large corporations. Although access to cloud services has become more convenient over time, this flexibility has made pricing more complicated. This complexity is why we have developed this model. Having a hedonic model will allow us to separate changes in the quality of the cloud service from changes in the price.

Several of these characteristics can be offered as standalone services or products that have significant impact on IaaS pricing and quality. We include storage as a characteristic because cloud service providers typically provide a small amount of storage with IaaS. They also sell stand-alone storage, but we are not including that in our data sets or models. Microprocessor hardware is also included as a price determining characteristic because the generation and model of a microprocessor can cause significant quality changes for cloud packages that may seem unchanged at face value. For instance, Diane Coyle and David Nguyen note the speed and changes in AWS packages between generations in their paper, "Cloud Computing and National Accounting." In November 2017, AWS claimed the new IaaS EC2 M5 instances would provide a 14% performance improvement over the previous M4 generation instances without a change in price. The only change announced between the two generations was an upgrade to a more modern microprocessor[4].

There are also several characteristics that can have an impact on price which are not quantifiable using publicly available data, and so, are not a focus of this model. For example, prices often differ by region because a higher concentration of larger data distribution centers in an area as well as the proximity of data distribution centers to the consumer have an effect on speed and performance of cloud services. This effect is referred to as latency. However, because cloud services are available to the consumer from anywhere, we have no way of knowing to what degree latency impacts overall instance performance. Likewise, there is no way for us to know how much consumers use these products over a certain time frame, so we choose to focus on time-based fees, specifically per hour costs of each instance.

Constructing PPI Cloud Services Hedonic Models

Hedonic regression models provide a means for quality adjusting services. The basic form of a hedonic model is a linear regression with price per hour being the dependent variable and characteristics of the service being the independent variables. The independent variables represent the different attributes that the service is composed of and the coefficients on those variables represent the cost of those attributes.

Previous work in applying hedonics to cloud computing exists. Zhang constructed a dataset for AWS and estimated hedonic models for 2009-2015[5]. These models were assessed on both multiple time periods (time dummy method) and single time periods (adjacent period method) to estimate quality adjusted measures of price change for AWS cloud services.

Mitropoulou et al also applied hedonics to cloud computing[6]. They assembled a dataset of 25 cloud service providers and estimated a hedonic model for a single time period. Coyle and Nguyen computed

[4] Diane Coyle and David Nguyen, "Cloud Computing and National Accounting," *ESCoE Discussion Paper 2018-2019.* December 2018.
[5] Liang Zhang, "Price trends for cloud computing services," *Honors Thesis Collection*, 386, 2016, https://repository.wellesley.edu/thesiscollection/386
[6] Persefoni Mitropoulou, Evangelia Filiopoulou, Stavroula Tsaroucha, Christos Michalakelis, and Mara Nikolaidou, "A Hedonic Price Index for Cloud Computing Services," *Proceedings of the 5th International Conference on Cloud Computing and Services Science,"* pages 499-505, 2015.

AWS prices per ECU and compared them to a nominal price index per product class, using the EC2 'large' and 'x.large' instances available in the UK from 2010Q1 to 2018Q3.[7]

For the models in this article, we estimated models using the time dummy hedonic method. The methodology we use is drawn from "The Rise of Cloud Computing: Minding Your P's, Q's and K's" by David Byrne, Carol Corrado, and Daniel Sichel[8]. We use time dummy hedonic models because they allow us to calculate price changes that occur from changes to several interrelated characteristics without having to be overly concerned about the magnitude of the coefficients for the characteristics, or the proportions of overall sales dedicated to each instance variation. Revenue and sales information at the product level are not often released to the public.

Because we use time dummy models, the dataset consists of two or more time periods of data. Specifically, we use overlapping two-quarter datasets. For example, the dataset for the first model consists of the second quarter and third quarter of 2017 and the dataset for the second model consists of the third quarter and fourth quarter of 2017.

In addition to variables representing characteristics of cloud computing services, the models also have a time dummy variable. The time dummy variable represents whether an observation is from the first quarter in the dataset or the second quarter. With differences in characteristics being accounted for by the other independent variables, the time dummy variable gives the quality adjusted price change between the two quarters. To use the estimate of price change from the model in a PPI index, we:

1. Exponentiate e to the time dummy coefficient from the model and then subtract the resulting value by one to represent quality adjusted price change in decimal form
2. Adjust prices for PPI items represented by the model so that those prices change by the amount calculated in step one

The model is intended to be used to directly estimate quality adjusted price change. Typically, a producer price index is comprised of items that represent specific services and their associated prices. In this case, items represent the aggregation of services included in the dataset used in the model. The

[7] Diane Coyle and David Nguyen, "Cloud Computing and National Accounting," *ESCoE Discussion Paper 2018-2019.* December 2018.
[8] David Byrne, Carol Corrado, and Daniel Sichel, "The Rise of Cloud Computing: Minding Your P's, Q's and K's," *Measuring and Accounting for Innovation in the 21st Century*.

item prices are being used to show quality adjusted price change calculated from the model as explained above.

The dataset is assembled to back-test quality adjustment models. It includes quarterly observations from the second quarter of 2017 through the second quarter of 2019 for the three largest companies in the industry: Amazon Web Services (AWS), Google Cloud, and Microsoft Azure.

The selection of characteristics to include in the model is important because it helps determine the magnitude of the time dummy variable, and thus the estimated price change. One of the key drivers of quality change with cloud services is the microprocessor used in the servers. Microprocessors undergo continual quality change and this change is responsible for improvements in a range of high tech goods and services from smart phones to artificial intelligence. Each cloud provider uses a limited number of microprocessor models in their servers. This limited number of models makes it difficult to have enough variation in microprocessor characteristics for the model to estimate significant coefficients.

AWS has a measure of the performance of the microprocessor used in their Elastic Compute Cloud (EC2) service called EC2 Compute Unit (ECU).[9] Because ECU is available for all AWS cloud services and it is calculated by AWS itself, ECU is a credible gauge of microprocessor performance. Neither Microsoft Azure nor Google Cloud have a microprocessor measure like AWS's ECU. Byrne, Corrado, and Sichel used AWS ECU in the models in their paper, and we have emulated this approach, which we will show later in this article[10].

There are third party microprocessor benchmarks, such as SPEC CPU and PassMark CPU benchmark, but both of these only have results for a limited number of microprocessors used in cloud services. This limited number of results is too small to support a cloud hedonic model.

Fortunately, characteristics information is available for all microprocessors. However, microprocessors are complicated devices, and selecting the characteristics to include in the model is challenging for two reasons[11]. First, only a few different microprocessors are used by any one cloud service provider which limits the number of microprocessor characteristics the model can support. As described by Sawyer and

---

[9] AWS FAQs, https://aws.amazon.com/ec2/faqs/.

[10] David Byrne, Carol Corrado, and Daniel Sichel, "The Rise of Cloud Computing: Minding Your P's, Q's and K's," *Measuring and Accounting for Innovation in the 21st Century*.

[11] David M. Byrne, Stephen D. Oliner, and Daniel E. Sichel, "How fast are semiconductor prices falling?" Review of Income and Wealth, vol. 64, no. 3, April 2017, pp. 679–702.

So in "A New Approach for Quality Adjusting PPI Microprocessors", the main characteristics of microprocessors are as follows[12]:

- cores – a hardware term that describes the number of independent central processing units (CPUs) on a single computing component (die or chip)
- threads – a software term for the basic ordered sequence of instructions that can be passed through or processed by a single CPU core
- thermal design power (TDP) – the average power, in watts, that the microprocessor dissipates when operating at base frequency with all cores active under an Intel-defined high-complexity workload
- base frequency – the rate at which the microprocessor's transistors open and close (The microprocessor base frequency is the operating point at which TDP is defined. Frequency is measured in gigahertz, or billions of cycles per second.)
- turbo frequency – the maximum single-core frequency at which the microprocessor is capable of operating using Intel Turbo Boost Technology
- cache – an area of fast memory located on the microprocessor (Intel's Smart Cache refers to the architecture that allows all cores to dynamically share access to the last level cache)

Second, cloud services are priced by virtual CPU (vCPU). Each vCPU corresponds to a microprocessor thread, which means that each vCPU is only using a part of the microprocessor. Consequently, each vCPU only uses part of the microprocessor cache and accounts for part of the TDP. The cache and TDP variables have to be multiplied by the proportion of threads (vCPUs) used by each individual cloud computing service to the total threads in the microprocessor. For example, if a cloud computing service has two vCPUs and the microprocessor used to provide the service has 10 threads, 8 MB of cache, and a TDP of 100 watts, then the cloud service uses 1.6 MB of cache and accounts for 20 watts of TDP. Base and turbo frequency are the same throughout the microprocessor so they do not need to be adjusted.

In "A New Approach for Quality Adjusting PPI Microprocessors", Sawyer and So used statistical learning techniques to select specifications for microprocessor hedonic models. These techniques, in turn, were taken from *An Introduction to Statistical Learning*[13]. Because many of the price determining

---

[12] Steven D. Sawyer and Alvin So, "A new approach for quality-adjusting PPI microprocessors," Monthly Labor Review, U.S. Bureau of Labor Statistics, December 2018, https://doi.org/10.21916/mlr.2018.29.

[13] G. James, D. Witten, T. Hastie and R. Tibshirani, *An Introduction to Statistical Learning: with Applications in R*, Springer Texts in Statistics 103, DOI 10.1007/978-1-4614-7_6, Springer Texts in Statistics+Business Media New York, 2013.

characteristics for cloud computing services are the same as for microprocessors, we applied the same statistical learning techniques that Sawyer and So used for microprocessors hedonic models.

The technique starts with prescreening the data. Our dataset for AWS has seven variables (not including the time dummy variables). We start with calculating the residual sum of squares (RSS) for every one regressor model. The one regressor model with the lowest RSS is selected as a prescreened model. We then repeat the procedure for all of the two regressor models. The two regressor model with the lowest RSS is selected as a prescreened model. We continue this process, increasing the number of regressors by one each time, until we have seven prescreened models. To select the best model from the prescreened models, we use repeated *k*-fold cross validation as explained in the following steps:

1. Split data set into *k* parts
2. Hold out one of the *k* parts and estimate the models on the remaining parts
3. Use models estimated in step 2 to predict prices for observations in *k* part
4. Square the difference between predicted prices and actual prices from the data set
5. Hold out each of the *k* parts in turn and repeat steps one through three
6. Repeat process multiple times but split the data differently each time
7. Take average of squared errors to calculate mean squared error (MSE)
8. Model with lowest MSE is selected

The selected models with ECU used to represent microprocessor performance are listed below. Please note, all of the models in this paper have log price as the dependent variable and use two adjacent quarters of data. The cloud services that continued from one quarter to the next never had any price changes. Only models that contained quarters with exiting or entering services had price change. We ran the statistical learning technique used in Sawyer and So on the quarters with no exit or entry to show the stability of the service characteristic coefficients.

AWS using ECU (17Q2-18Q2)

	17Q2-17Q3	17Q3-17Q4	17Q4-18Q1	18Q1-18Q2
Quarter dummy	0	0	-0.0296	0
	(0.0455)	(0.0455)	(0.0435)	(0.0372)
Log (VCPU)	-1.6094*	-1.6094*		
	(0.4586)	(0.4586)		
Log (memory)	0.7051*	0.7051*	0.5379*	0.5376*
	(0.0792)	(0.0792)	(0.0446)	(0.0423)
Log (storage)	0.1401*	0.1401*	0.1608*	0.1588*
	(0.0224)	(0.0224)	(0.0229)	(0.0232)
SSD	-1.1939*	-1.1939*	-1.2873*	-1.2506*
	(0.183)	(0.183)	(0.1911)	(0.1931)
Log (ECU)	1.9327*	1.9327*	0.4407*	0.4454*
	(0.4124)	(0.4124)	(0.0511)	(0.0468)
Windows	0.4676*	0.4676*	0.5047*	0.5318*
	(0.0455)	(0.0455)	(0.0415)	(0.0372)
Observations	128	128	152	176
AdjR2	0.965	0.965	0.9659	0.9683

*Significant at the 5-percent level

AWS using ECU (18Q2-19Q2)

	18Q2-18Q3	18Q3-18Q4	18Q4-19Q1	19Q1-19Q2
Quarter dummy	0	0.0671	0	0
	(0.0372)	(0.0478)	(0.0479)	(0.0479)
Log (VCPU)		1.0276*	1.3102*	1.3102*
		(0.2099)	(0.1659)	(0.1659)
Log (memory)	0.5376*	0.4427*	0.452*	0.452*
	(0.0423)	(0.0436)	(0.036)	(0.036)
Log (storage)	0.1588*	0.1086*	0.0634*	0.0634*
	(0.0232)	(0.0238)	(0.0244)	(0.0244)
SSD	-1.2506*	-0.8662*	-0.5449*	-0.5449*
	(0.1931)	(0.1801)	(0.1687)	(0.1687)
Log (ECU)	0.4454*	-0.5007*	-0.7908*	-0.7908*
	(0.0468)	(0.2005)	(0.1709)	(0.1709)
Windows	0.5318*	0.5183*	0.5083*	0.5083*
	(0.0372)	(0.0457)	(0.0479)	(0.0479)
Observations	176	206	236	236
AdjR2	0.9683	0.9466	0.9349	0.9349

*Significant at the 5-percent level

Only the models for 17Q4-18Q1 and 18Q3-18Q4 had price changes. Prices are stable, which means price changes are caused by the entry and exit of cloud services. Some of the variables have counterintuitive signs on their coefficients in some of the models, such as Log (ECU) in the last three models. This phenomenon can arise when variables are correlated with each other. With time dummy models, we are mainly interested in the time dummy coefficient.

Even though most variables are being selected, there is still value in using the statistical learning specification algorithm. For instance, Log (vCPU) is not selected for three of the models, and this variable is one of the main price determining characteristics of cloud services. Without the statistical learning specification algorithm, we would have not known that omitting Log (vCPU) would produce a model with better performance in those three models.

We also estimate models for AWS using microprocessor characteristics instead of ECU. The ECU models serve as a benchmark we can use to measure the performance of the characteristics models. This measure of performance will be useful for gauging the appropriateness of using microprocessor characteristics in the Microsoft Azure and Google Cloud models where an ECU-like variable is not available.

For the characteristics models, the vCPU variable is omitted because it is strongly correlated with cache and TDP. The amount of cache or TDP used by a cloud service is proportional to the number of vCPUs, which was explained previously. Because the base and turbo frequency variables are closely correlated and there are so few microprocessors in the data set, model selection using the Sawyer and So technique was done twice, once using base frequency and omitting turbo frequency and once using turbo frequency and omitting base frequency.

AWS using CPU Characteristics with Base Frequency (17Q2-18Q2)

	17Q2-17Q3	**17Q3-17Q4**	**17Q4-18Q1**	**18Q1-18Q2**
Quarter dummy	0	0	-0.0358	0
	(0.0338)	(0.0338)	(0.0368)	(0.0305)
Log (memory)	0.999*	0.999*	0.8248*	0.8302*
	(0.069)	(0.069)	(0.0597)	(0.0648)
Log (storage)	0.0521	0.0521	0.1595*	0.1696*
	(0.0285)	(0.0285)	(0.0237)	(0.025)
SSD	-0.5403*	-0.5403*	-1.3028*	-1.3855*
	(0.2419)	(0.2419)	(0.1969)	(0.2135)
Log (base frequency)	1.5521*	1.5521*	2.7659*	3.1469*
	(0.3745)	(0.3745)	(0.2894)	(0.4642)
Log (cache)	-2.503*	-2.503*	0.1327*	0.4323*
	(0.3498)	(0.3498)	(0.0598)	(0.1015)
Log (TDP)	2.4708*	2.4708*		-0.3015*
	(0.3265)	(0.3265)		(0.1429)
Windows	0.4676*	0.4676*	0.5047*	0.5318*
	(0.0338)	(0.0338)	(0.0345)	(0.0305)
Observations	128	128	152	176
AdjR2	0.9807	0.9807	0.9764	0.9786

*Significant at the 5-percent level

AWS using CPU Characteristics with Base Frequency (18Q2-19Q2)

	18Q2-18Q3	**18Q3-18Q4**	**18Q4-19Q1**	**19Q1-19Q2**
Quarter dummy	0	0.0708	0	0
	(0.0305)	(0.0431)	(0.0456)	(0.0456)
Log (memory)	0.8302*	0.7399*	0.7052*	0.7052*
	(0.0648)	(0.057)	(0.0542)	(0.0542)
Log (storage)	0.1696*	0.1066*	0.0598*	0.0598*
	(0.025)	(0.0229)	(0.0231)	(0.0231)
SSD	-1.3855*	-0.9008*	-0.5604*	-0.5604*
	(0.2135)	(0.1753)	(0.1591)	(0.1591)
Log (base frequency)	3.1469*	2.5898*	2.1997*	2.1997*
	(0.4642)	(0.4648)	(0.4955)	(0.4955)
Log (cache)	0.4323*	0.6801*	0.7981*	0.7981*
	(0.1015)	(0.1401)	(0.1412)	(0.1412)
Log (TDP)	-0.3015*	-0.4262*	-0.4857*	-0.4857*
	(0.1429)	(0.1717)	(0.1777)	(0.1777)
Windows	0.5318*	0.5183*	0.5083*	0.5083*
	(0.0305)	(0.0423)	(0.0456)	(0.0456)
Observations	176	206	236	236
AdjR2	0.9786	0.9544	0.9411	0.9411

*Significant at the 5-percent level

The statistical learning technique is selecting most of the variables for the models. These models are also showing price changes for 17Q4-18Q1 and 18Q3-18Q4, just as the models using ECU did. The price changes are somewhat larger in the models using characteristics, but they are not drastically different.

AWS using CPU Characteristics with Turbo Frequency (17Q2-18Q2)

	17Q2-17Q3	**17Q3-17Q4**	**17Q4-18Q1**	**18Q1-18Q2**
Quarter dummy	0	0	-0.0422	0
	(0.0349)	(0.0349)	(0.043)	(0.034)
Log (memory)	0.949*	0.949*	0.596*	0.5655*
	(0.0577)	(0.0577)	(0.0509)	(0.0438)
Log (storage)			0.1341*	0.1391*
			(0.0222)	(0.0227)
SSD	-0.1351*	-0.1351*	-1.0837*	-1.1101*
	(0.0499)	(0.0499)	(0.1859)	(0.1877)
Log (turbo frequency)				
Log (cache)	-3.4095*	-3.4095*	-0.3434*	-0.1319
	(0.2746)	(0.2746)	(0.1356)	(0.0925)
Log (TDP)	3.4404*	3.4404*	0.7166*	0.5407*
	(0.2275)	(0.2275)	(0.1059)	(0.0701)
Windows	0.4676*	0.4676*	0.5047*	0.5318*
	(0.0349)	(0.0349)	(0.0383)	(0.034)
Observations	128	128	152	176
AdjR2	0.9794	0.9794	0.9708	0.9735

*Significant at the 5-percent level

AWS using CPU Characteristics with Turbo Frequency (18Q2-19Q2)

	18Q2-18Q3	**18Q3-18Q4**	**18Q4-19Q1**	**19Q1-19Q2**
Quarter dummy	0	0.083	0	0
	(0.0341)	(0.0453)	(0.046)	(0.046)
Log (memory)	0.5373*	0.5516*	0.5676*	0.5676*
	(0.0345)	(0.0317)	(0.0276)	(0.0276)
Log (storage)	0.1423*	0.0516*		
	(0.0229)	(0.022)		
SSD	-1.1375*	-0.4544*	-0.0936	-0.0936
	(0.1888)	(0.1714)	(0.0514)	(0.0514)
Log (turbo frequency)		-1.8409*	-1.9731*	-1.9731*
		(0.6074)	(0.5748)	(0.5748)
Log (cache)		-0.4847	-0.3937	-0.3937
		(0.2634)	(0.2882)	(0.2882)
Log (TDP)	0.4381*	0.944*	0.8612*	0.8612*
	(0.0349)	(0.2557)	(0.2772)	(0.2772)
Windows	0.5318*	0.5183*	0.5083*	0.5083*
	(0.0341)	(0.0432)	(0.046)	(0.046)
Observations	176	206	236	236
AdjR2	0.9733	0.9524	0.9401	0.9401

*Significant at the 5-percent level

Again, as with the characteristics models using base frequency, the characteristics models using turbo frequency show price changes for 17Q4-18Q1 and 18Q3-18Q4. But the characteristics models using turbo frequency show a larger deviance from the ECU models than the characteristics models using base frequency. Unlike base frequency, which was selected for all models, turbo frequency was only selected in three of the eight models. Overall, the characteristics models using base frequency show better performance than characteristics models using turbo frequency.

As of 2021, AWS no longer publishes ECU for its cloud computing services. Fortunately, as this paper has shown, microprocessor characteristics can be used in place of ECU if BLS decides to use hedonic models in the PPI for Hosting, ASP, and other IT infrastructure.

We use the statistical learning technique used by Sawyer and So to estimate models for Microsoft Azure separately for base frequency and turbo frequency.

Microsoft Azure using CPU Characteristics with Base Frequency (17Q2-18Q2)

	17Q2-17Q3	**17Q3-17Q4**	**17Q4-18Q1**	**18Q1-18Q2**
Quarter dummy	0	-0.0079	0.0095	0
	(0.0284)	(0.0279)	(0.0208)	(0.0197)
Log (memory)	0.4646*	0.4885*	0.4791*	0.4315*
	(0.0244)	(0.0245)	(0.0311)	(0.0155)
Log (storage)	0.1273*	0.1162*	0.1231*	0.1508*
	(0.0165)	(0.0165)	(0.0193)	(0.0115)
SSD	0.4903*	0.5135*	0.5155*	0.4918*
	(0.0252)	(0.0243)	(0.0259)	(0.0251)
Log (base frequency)		-0.6297	-0.9335*	-0.7991*
		(0.3723)	(0.1297)	(0.1036)
Log (cache)	0.8013*	0.5283*	0.4003*	0.4039*
	(0.061)	(0.1492)	(0.0262)	(0.0243)
Log (TDP)	-0.3721*	-0.1176		
	(0.0664)	(0.154)		
Windows	0.4055*	0.422*	0.4436*	0.4503*
	(0.0293)	(0.028)	(0.024)	(0.021)
Observations	136	148	150	140
AdjR2	0.9866	0.9872	0.9893	0.9919

*Significant at the 5-percent level

Microsoft Azure using CPU Characteristics with Base Frequency (18Q2-19Q2)

	18Q2-18Q3	**18Q3-18Q4**	**18Q4-19Q1**	**19Q1-19Q2**
Quarter dummy	0.0295	0.0049	-0.0018	-0.0036
	(0.0238)	(0.0168)	(0.0155)	(0.0157)
Log (memory)	0.3166*	0.2625*	0.2631*	0.2584*
	(0.0225)	(0.0098)	(0.0095)	(0.0099)
Log (storage)	0.2569*	0.3493*	0.3487*	0.3544*
	(0.0213)	(0.0162)	(0.017)	(0.0171)
SSD	0.2625*	0.0785*	0.0774*	0.0674*
	(0.0316)	(0.0138)	(0.0137)	(0.0144)
Log (base frequency)	-2.9143*	-4.5942*	-4.7023*	-4.7626*
	(0.3972)	(0.3076)	(0.3054)	(0.3081)
Log (cache)	-0.6378*	-1.3068*	-1.3475*	-1.379*
	(0.1618)	(0.0958)	(0.0933)	(0.0943)
Log (TDP)	1.0623*	1.7011*	1.7406*	1.7691*
	(0.1607)	(0.0834)	(0.0779)	(0.0788)
Windows	0.4415*	0.4508*	0.4683*	0.4701*
	(0.0277)	(0.017)	(0.0161)	(0.0164)
Observations	132	138	152	152
AdjR2	0.9873	0.9951	0.9951	0.9949

*Significant at the 5-percent level

Microsoft Azure using CPU Characteristics with Turbo Frequency (17Q2-18Q2)

	17Q2-17Q3	17Q3-17Q4	17Q4-18Q1	18Q1-18Q2
Quarter dummy	0	-0.0024	0.005	0
	(0.0284)	(0.0283)	(0.0204)	(0.0197)
Log (memory)	0.4646*	0.4814*	0.4659*	0.4308*
	(0.0244)	(0.0257)	(0.0274)	(0.0157)
Log (storage)	0.1273*	0.1199*	0.1307*	0.1509*
	(0.0165)	(0.0171)	(0.0171)	(0.0115)
SSD	0.4903*	0.5057*	0.5059*	0.4894*
	(0.0252)	(0.0242)	(0.0263)	(0.0261)
Log (turbo frequency)		-1.276*	-1.1989*	-1.0693*
		(0.4215)	(0.3074)	(0.2869)
Log (cache)	0.8013*	0.6062*	0.6136*	0.5848*
	(0.061)	(0.0721)	(0.0556)	(0.043)
Log (TDP)	-0.3721*	-0.1892*	-0.2075*	-0.1787*
	(0.0664)	(0.0801)	(0.067)	(0.0517)
Windows	0.4055*	0.4231*	0.4437*	0.4507*
	(0.0293)	(0.0275)	(0.0237)	(0.0209)
Observations	136	148	150	140
AdjR2	0.9866	0.9873	0.9895	0.9919

*Significant at the 5-percent level

Microsoft Azure using CPU Characteristics with Turbo Frequency (18Q2-19Q2)

	18Q2-18Q3	18Q3-18Q4	18Q4-19Q1	19Q1-19Q2
Quarter dummy	0.0383	0.0441	-0.0018	-0.0036
	(0.024)	(0.0273)	(0.0269)	(0.0273)
Log (memory)	0.3266*	0.2633*	0.2633*	0.2587*
	(0.0223)	(0.0189)	(0.0203)	(0.0206)
Log (storage)	0.253*	0.2806*	0.2681*	0.2726*
	(0.0214)	(0.0239)	(0.0275)	(0.0279)
SSD	0.2515*	0.1465*	0.1569*	0.1481*
	(0.0295)	(0.0242)	(0.0265)	(0.027)
Log (turbo frequency)	-3.5819*	-1.515*	-0.9662*	-0.9857*
	(0.3235)	(0.3951)	(0.2928)	(0.2934)
Log (cache)		-0.2078*	-0.2266*	-0.2445*
		(0.0742)	(0.0815)	(0.0826)
Log (TDP)	0.4197*	0.662*	0.6913*	0.7073*
	(0.0273)	(0.0639)	(0.0653)	(0.0659)
Windows	0.4379*	0.4052*	0.4198*	0.421*
	(0.0269)	(0.0247)	(0.0247)	(0.025)
Observations	132	138	152	152
AdjR2	0.9871	0.9859	0.985	0.9846

*Significant at the 5-percent level

Unlike AWS, there is no major difference between the models using base frequency and the models using turbo frequency. Both sets of models have similar time dummy variables, and the magnitude and sign of base frequency and turbo frequency are similar in the respective models.

For Google Cloud, all of the cloud services in a given region use the same microprocessors. We constructed our dataset from two different regions to provide a mix of microprocessors. Over the time period of 17Q2 to 19Q1, there were no changes in products or prices. With this lack of change, there would of course be no price change for a hedonic model to capture. In the second quarter of 2019, there was a change in the frequency of the microprocessors in one of the regions. We used the statistical learning algorithm with both frequency variables, but in both cases the frequency variables were not selected. Because we know exactly what changed with the cloud services, we estimated models with both frequency variables to see if they yielded any appreciable quality adjusted price change. There was a strong correlation between the region variables and the frequency variables, so the region variables were omitted. Likewise, there was a strong correlation between cache and TDP, so TDP was omitted. Below are results for the model using base frequency and turbo frequency.

Google Cloud using CPU Characteristics with Base Frequency (19Q1-19Q2)

	19Q1-19Q2
Quarter dummy	-0.0011
	(0.0174)
Log (memory)	0.1850*
	(0.0087)
Log (base frequency)	-0.0413
	(0.3790)
Log (cache)	0.8172*
	(0.0102)
Windows	0.5034*
	(0.0142)
Observations	152
AdjR2	0.9956

*Significant at the 5-percent level

Google Cloud using CPU Characteristics with Turbo Frequency (19Q1-19Q2)

	19Q1-19Q2
Quarter dummy	0.0003
	(0.0143)
Log (memory)	0.1850*
	(0.0087)
Log (turbo frequency)	-0.0133
	(0.0873)
Log (cache)	0.8172*
	(0.0102)
Windows	0.5034*
	(0.0142)
Observations	152
AdjR2	0.9956

*Significant at the 5-percent level

The models are remarkably similar except for the quarter dummy and frequency coefficients. This similarity suggests that the change in microprocessor frequency in the second quarter of 2019 caused negligible quality adjusted price change and it helps illustrate why the statistical learning algorithm did not select either frequency variable.

Conclusion

Our results show that a time dummy hedonic model is able to estimate quality adjusted price change for cloud computing services. One of the key drivers of quality change for cloud services is technological improvements in microprocessors. Being able to demonstrate that models using microprocessor characteristics are able to produce similar results as models using AWS's ECU measure is important for validating the use of microprocessor characteristics in cloud services models. We have also applied the statistical learning method used in the Sawyer and So microprocessors paper. This method of specification selection allows the models to change over time as cloud services change. It also makes our method of choosing a specification transparent to our data users which should bolster the perceived integrity of the cloud models. With a hedonic model for cloud services, BLS will have another method available for PPI estimates that accounts for quality change in an industry that experiences rapid technological progress and has become crucial for the information technology sector.