Modeling Multi-Attribute Demand for Sustainable Cloud Computing With Copulae

Maryam Ghasemi
Computer Science Department
Boston University
ghasemi@bu.edu

Benjamin Lubin
School of Management
Boston University
blubin@bu.edu

Abstract

As cloud computing gains in popularity, understanding the patterns and structure of its loads is increasingly important in order to drive effective resource allocation, scheduling and pricing decisions. These efficiency increases are then associated with a reduction in the data center environmental footprint. Existing models have only treated a single resource type, such as CPU, or memory, at a time. We offer a sophisticated machine learning approach to capture the joint-distribution. We capture the relationship among multiple resources by carefully fitting both the marginal distributions of each resource type as well as the non-linear structure of their correlation via a copula distribution. We investigate several choices for both models by studying a public data set of Google datacenter usage. We show the Burr XII distribution to be a particularly effective choice for modeling the marginals and the Frank copula to be the best choice for stitching these together into a joint distribution. Our approach offers a significant fidelity improvement and generalizes directly to higher dimensions. In use, this improvement will translate directly to reductions in energy consumption.

1 Introduction

In 2013, US data centers used about 91 billion kWh [NRDC, 2013], showing a 50% increase since 2006. This number is 2.2% of the 4 trillion kWh consumed in total, and is equal to the amount of energy used by 5.6 million average US households [EIA, 2015]. This consumption growth is expected to continue, with another 50% increase in the next five years [NRDC, 2013]. This will require construction of an additional 16 500 MW power plants, and result in 100 million metric tons of CO_2 pollution per year. While the need for such facilities will not abate, much can be done to reduce the energy required. Recent work has shown that considerable energy can be saved by using heterogeneous processor types in data centers [Guevara *et al.*, 2014], and then assigning processes to be run on the most energy-efficient resources for handling them by sophisticated resource allocation methods [Lubin *et al.*, 2009; Beloglazov and Buyya, 2010; Beloglazov *et al.*, 2012].

However, such methods are only as good as the demand models that drive them, and the models in the literature handle only single resource types, not the full joint distribution of demand. Having high fidelity demand models is critical if the enormous rise in data center energy usage is to be curtailed.

Analyzing the characteristics of workloads is a long-studied problem with application to load balancing, system availability, failure, and process behavior. The literature covering this topic originates in the 1960s. In some of the earliest such studies, Rosin [1965] and Walter and Wallace [1967] study data for more than $10,000$ jobs run at the University of Michigan Computing Center. They find that the shape of CPU process time is skewed and used a simple exponential distribution to model it. Such work continued into the 1970s, for example Agrawala *et al.* [1976] presented three simple models of system workload. A large literature in the high performance computing literature has also addressed demand modeling [Li *et al.*, 2004].

Most of this work has focused on one specific attribute of demand (e.g., CPU time, memory usage, or I/O operations). To the best of our knowledge there is no existing work on the joint distribution of resource usage. We believe that there is important structure in the correlation between different attributes of demand, and that therefore an understanding the nature of the joint distribution will significantly improve allocation, scheduling and pricing algorithms among other uses for such models, which in turn will yield a significant positive environmental impact due to reduced energy usage.

Accordingly, in this paper, we develop a multivariate model for computational resource usage. Moreover, as part of our model we propose to use a more flexible distribution, the Burr XII, as our model of univariate demand. We will show that this model has significantly more power to capture real-world usage patterns than the models used to date in the literature. Having modeled the individual attributes, we then combine them into a multi-attribute model through the machinery of copulae [Sklar, 1959].

Copulae are an elegant way to build and fit complex joint distributions that have been used to great effect in finance, but which have not been used in computational demand modeling. By employing this method we have been able to accurately handle an otherwise-intractable modeling problem. In

this work, we focus on the bivariate case, specifically capturing the joint demand for CPU and memory, typically the two most important resource types. We then validate our approach based on a public data set from Google [Reiss *et al.*, 2011] that contains 29 days of workload data from a large production cluster. Using this data, we examine a large number of models from the family we propose, and demonstrate the effectiveness of a particular model that combines Burr XII marginals with the Frank copulae. The proposed model can be directly generalized to more resource types, though we are limited by our data in doing so here. In this paper we use machine learning and computational statistics to model data center usage, offering the following contributions:

- A novel application of the Burr XII distribution to modeling the distributions of individual workload attributes (Section 2.2).

- The first demand models that capture the joint distribution over multiple attribute types, built by applying copula methods to our marginal models (Section 2.3).

- An application of these new tools to a large Google data set (Section 3).

- Validation of this approach by employing a parametric bootstrapped Kolmogorov-Smirnov test on both the univariate and multi-variate data (Section 5)

2 Theory and Models

Here we provide the machinery needed to create multi-attribute models and apply them to our setting.

2.1 Preliminaries

Given a set of size d of correlated random variables $X_i, \forall i \in \mathcal{D} = \{1, \dots, d\}$, we define the joint cumulative distribution function (CDF) as $F(\vec{x})$, as the probability that $X_i \leq x_i \, \forall i \in \mathcal{D}$ where $\vec{x} \in \mathbb{R}^d$. We then define the joint probability density function (PDF) in the usual way as:

$$f(\vec{x}) = \left. \frac{\partial F}{\partial X_1 \dots \partial X_d} \right|_{\vec{x}}$$

The *marginal* density function for variable X_i is then simply its distribution without regard to all the other variables:

$$f_i(x_i) = \int_{\vec{x}_{j<i}} \dots \int_{\vec{x}_{k>i}} f(\langle \vec{x}_j; x_i; \vec{x}_k \rangle) d\vec{x}_j d\vec{x}_k$$

Where $\langle \cdot ; \cdot \rangle$ represents vector composition. Here the integration ranges over all the dimensions but i, the marginal being specified. We will denote the corresponding CDF as $F_i(\vec{X})$.

Complex multivariate distributions are notoriously hard to model and fit. Consequently, we here appeal to the powerful formalism of *copulae*. Copulae enable the construction of a model that captures the various marginal distributions of a empirical data *separately* from the "coupling" of these marginals. As we shall see, this separation makes both forming and fitting a complex model far more tractable.

The word *copula* was first introduced by Sklar [1959] in a theorem that described the necessary properties of functions that "join together" one dimensional marginal distributions to

Archimedean Copula	Generator Function ψ
Gumbel Copula	$\psi(u_i) = \exp\left(-\left[\sum_i^n \log^\lambda(\frac{1}{u_i})\right]^{1/\lambda}\right)$
Frank Copula	$\psi(u_i) = \log_\lambda\left(\frac{\prod_i^n(\lambda^{u_i}-1)}{(\lambda-1)^{n-1}}+1\right)$
Clayton Copula	$\psi(u_i) = \left(\sum_i^n u_i^{1/\lambda} - 1\right)^{-\lambda}$

Table 1: Several Archimedean copulae and their formulae

create multivariate distributions. Concretely, a copula function C is a multivariate distribution defined on the unit hypercube $[0, 1]^d$ with uniform marginals:

$$C(u_1, u_2, \dots, u_d) = Pr[U_1 \leq u_1, U_2 \leq u_2, \dots, U_d \leq u_d]$$

where $U_i \sim \text{Uniform}(0, 1), i \in \mathcal{D}$, along with the boundary conditions that:

$$[u_i = 0] \Rightarrow [C(u_1, \dots, u_i, \dots, u_d) = 0] \, \forall i \in \mathcal{D} \wedge u_{\neq i} \in \mathbb{R}$$
$$[u_{\neq i} = 1] \Rightarrow [C(u_1, \dots, u_i, \dots, u_d) = u_i] \, \forall i \in \mathcal{D} \wedge u_i \in \mathbb{R}$$

Furthermore, according to Sklar's theorem [Sklar, 1959], if CDF $F(\vec{x})$ is a continuous multivariate distribution, a unique copula function $C : [0, 1]^d \to [0, 1]$ exists such that

$$F(\vec{x}) = C(F_1(x_1), F_2(x_2), \dots, F_d(x_d))$$

for continuous marginal distributions $F_i(x_i)$. Consequently, copula C links the marginal distributions together to create a multivariate distribution. In theory, such a copula function can be arbitrarily complicated. We seek a simple parametric copula that fits our data well. Next, we describe several such copulae from literature that we evaluate in Section 5.2

The Gaussian copula, which is based upon the multivariate Normal distribution, is one such choice for modeling dependence. The Gaussian copula is defined as:

$$C_\Phi(u_1, u_2, \dots, u_d) = \Phi(F_1^{-1}(u_1), F_2^{-1}(u_2), \dots, F_d^{-1}(u_d), \Sigma)$$

where Φ is multivariate Normal CDF, Σ is a correlation matrix for the different variables, and the $F_1, F_2, \dots, F_d$ are arbitrary but well-defined marginal distributions.

Another important class of copula functions are the Archimedean copulae. Archimedean copulae have simple parametric form that allows for a variety of different dependence structures. The general form of an Archimedean copula is

$$C_A(u_1, u_2, \dots, u_d) = \psi^{-1}(\psi(u_1) + \psi(u_2) + \dots + \psi(u_d))$$

where ψ is a decreasing univariate function known as *generator* of the copula, carefully chosen such that C_A will obey the copula properties identified above. The Clayton, Gumbel and Frank copulae are each defined by a specific choice of Archimedean generator, as provided in Table 1, where λ is a free parameter that is available for fitting.

Joint dependence structure has been modeled using copulae in many fields, including reliability engineering, finance, and risk management. Yet, these powerful copula-based methods are unfamiliar to many researchers, and there is little literature comparing the efficacy of copulae in real settings.

2.2 Modeling the Marginal

Before building our full multivariate model, we must first ensure that we have accurately described the marginals in our domain, i.e. that we have appropriate univariate models for each of the demand attributes. Therefore, we next consider several distributions of demand for specific resource types including a new one not appearing elsewhere in the literature.

Process CPU usage (or runtime on a fixed processor type) is typically a key feature of computational loads. Early studies show that the general shape of the process runtime distribution is skewed [Walter and Wallace, 1967; Rosin, 1965], i.e., there are a lot of short jobs and a few long jobs. These papers use a simple exponential PDF to model process lifetime: $f(x) = \frac{1}{\theta}e^{x/\theta}$, where θ is a *scale parameter*. Later Leland and Ott [1986] and Harchol-Balter and Downey [1997] showed that process life distributions are actually heavy-tailed and better modeled by a Pareto distribution, with PDF function $f(x) = \xi x_m^{\xi}/x^{\xi+1}$ where x_m is location parameter and ξ is a shape parameter. There are also many studies of memory usage and its linear correlation with respect to CPU usage. For example, Parsons and Sevcik [1996], as well as Li *et al.* [2004] studied workload characteristics, including job size and memory usage. Chiang and Vernon [2001] found that most jobs have a very small memory usage per processor (under 32M) while 5% of jobs have a memory usage greater than 1 GB per processor, and a significant correlation between requested memory and number of processors. Work of this style describes memory usage statically; there has not been to our knowledge a distributional model for memory propsed. Moreover, none of these studies goes beyond simple linear correlation to model the full joint distribution over demand attributes as we do.

In our work, we have found empirically that the distribution of CPU usage (and memory) are not always decreasing, and a skewed bell-type structure is in fact possible. When confronted with such a bell shape, the first distributions one typically considers are the Normal and Log-Normal distributions. However, these are exponential family distributions which will not fit heavy-tailed data well. Because we believe our data may be heavy tailed, we additionally consider a somewhat more esoteric distribution, the Burr XII [Burr, 1942], also known as the Singh-Maddala distribution. The Burr XII distribution can fit a wide range of empirical data, as different values of its parameters cover a broad class of skewness and kurtosis. Hence, it is used in various fields such as finance, hydrology, and reliability to model a variety of data types. We choose it as a target because it can take on both a decreasing and a bell shape; it also generalizes the Gamma, Generalized Pareto and the Log-Logistic distributions which would otherwise be candidates in their own right [Burr, 1968; Tadikamalla, 1980].

The three parameter Burr XII PDF is given by

$$f(x) = \frac{\xi k (x/\beta)^{\xi-1}}{\beta((x/\beta)^{\xi}+1)^{k+1}}$$

and CDF

$$F(x) = 1 - (1 + (x/\beta)^{\xi})^{-k}$$

where $\xi > 0$ and $k > 0$ are shape parameters and $\beta > 0$ is the scale parameter. Because of its flexibility and appropriate properties, we will consider the Burr XII distribution both for CPU and memory marginals.

2.3 Modeling the Joint Distribution

Having modeled a single attribute like CPU runtime or memory usage by an appropriate marginal distribution we now turn to handling the joint distribution of multiple attributes. At present, we focus on a joint distribution of CPU and memory because this is what is in our data, although a generalization to further dimensions is immediate.

We first model the marginals as in the previous section. Then, we perform a *probability integral transform* on the data along each marginal. This converts a random variable A to a new random variable B, which is uniformly distributed on $[0, 1]$ by applying the CDF of A, F_A to itself: $B = F_A(A)$. Concretely, this means we apply each of our fitted marginal distributions $F_i \sim$ Burr XII to the corresponding dimension of our data. This procedure produces a distribution on the unit square with uniform margins but some more complex joint structure. We then seek a parametric model for this joint-uniform distribution: i.e., a copula as defined in Section 2.1.

By compositing our chosen copula with the marginal models, we then obtain a model of the full joint distribution that has the appropriate marginal distributions exactly, and where the joint dependence of the model is represented solely in the copula. This enables us to both define and fit a complete joint distribution in a principled manner.

3 The Data

Large open data sets for memory and CPU load are hard to come by, with the notable exception of the public Google workload traces we employ [Reiss *et al.*, 2011]. The traces are compromised of $11,000$ machines' logs from 29 days in May 2011. We restrict the set to the 54.7% of jobs that completed successfully, resulting in about $760,000$ processes. Reported CPU time is in microseconds, and memory usage is scaled between 0 and 10^7. We interpret a process' memory usage as the maximum observed over its runtime.

Before analyzing the data set some preparation is required. First, processes with 0 usage are omitted, which is a further 5% of the data. Further, while the bulk of the data comes from a reasonable spread of job types, the data includes a set of extreme outliers that are different in kind to the balance of the data, and thus the raw empirical distribution is multimodal in the tail. We believe this is a limitation of the data size (large though it is), and that if the number of job types included in the data were larger, these modes in the tail would disappear. To avoid this confounding our results, we restrict ourselves to the bulk of the jobs outside of these extreme outliers, or specifically to CPU usage below $5*10^6$ and memory usage below $7.5*10^4$, which is 82% of the raw data. Next, to interpolate between the job types (modes) in the data, we perform a bivariate kernel-density smoothing with a Normal kernel and a bandwidth of 0.4 and 0.3 for memory and CPU respectively.

4 Bootstrap Goodness-of-Fit Testing

Algorithm 1 The Bootstrap Kolmogorov-Smirnov goodness-of-fit test for a hypothesized family of distributions against observed data.

Input: b, Observed Data O, Hypothesized Distribution H
Output: p-value

1: $c \leftarrow 0$
2: $n \leftarrow$ Size of O
3: $\gamma_O \leftarrow$ Estimate parameters of H on O by MLE
4: $K_o \leftarrow D_{KS}(O, H(\gamma_O))$
5: **for** $i = 1 \ldots b$ **do**
6:     $R_i \leftarrow$ Generate s random points $\sim H(\gamma_o)$
7:     $\gamma_i \leftarrow$ Estimate Parameters of H on R_i by MLE
8:     $K_i \leftarrow D_{KS}(R_i, H(\gamma_i))$
9:     **if** $K_i \geq K_o$ **then**
10:         $c \leftarrow c + 1$
11:     **end if**
12: **end for**
13: p-value $\leftarrow c/n$
14: **return** p-value

Distribution	CPU Parameters	p value	Memory Parameters	p value
Burr XII	$\xi = 178000$ $c = .9$ $\kappa = .15$	.52	$\xi = 15900$ $c = 1.8$ $\kappa = 1.05$	.74
Generalized Pareto	$\xi = 839262$ $\theta = 0$ $\kappa = .26$	0	$\xi = 23540$ $\theta = 0$ $\kappa = -0.26$	0
Log-Normal	$\mu = 13.05$ $\sigma = 1.59$	0	$\mu = 13.16$ $\sigma = .91$	0
Log-Logistic	$\mu = 14.2$ $\sigma = 1.8$	0	$\mu = 9.56$ $\sigma = .5$	0

Table 2: Maximum Likelihood Estimate (MLE) parameters for CPU and Memory models and their associated p-values calculated by parametric bootstrapping.

Given an observed data set and a family of distributions that we hypothesize the data to be drawn from, we would like to know whether our hypothesis is a reasonable one, given the data. A standard approach to answering this kind of question is to use a goodness-of-fit test, which quantifies the plausibility of the hypothesis. In standard *statistical hypothesis testing*, one calculates a test statistic p, that captures the probability of the data being at least as "extreme" as that observed, given that the hypothesis is true. When this probability is less than a specified significance level, α, we reject the hypothesis. In typical usage, the hypothesis is a so called *null* (e.g. the data is explained by randomness), and so a rejection of this hypothesis in favor of an alternative and mutually exclusive one held by the researcher is the desired outcome, in that it provides evidence in favor of the alternative hypothesis. Thus, concretely, for disproving a null hypothesis you want $p < \alpha$. Here though, in goodness-of-fit testing, we wish to find evidence that our hypothesis *is* true (i.e., the data is consistent with our fitted distribution). Consequently, a small p-value (i.e., below α) is evidence *against* our proposed model, and we want $p > \alpha$, ideally significantly so [D'Agostino, 1986].

To perform such a test, it remains to choose the notion of "extreme" that defines p. Many such test statistics have been proposed. In this work, we employ the commonly used Kolmogorov-Smirnov (KS) statistic: $D_{KS}(O, F) = sup_x\{F_O(x) - F(x)\}$ where $F_O(x)$ is the empirical CDF of the observed data calculated as: $F_O(x) = \frac{1}{n}\sum_i^n \boldsymbol{I}(X_i \leq x)$ where $\boldsymbol{I}$ is the indicator function which is 1 when the given condition is true and 0 otherwise. D_{KS} measures the maximum point-wise difference in the CDF of the hypothesized distribution and the observed data.

When the hypothesis is that the data is drawn from a concrete distribution, not a parametrized family, then a simple goodness-of-fit test is available: One finds the quantile of $\sqrt{n}D_{KS}$ in the well-known *Kolmogorov* distribution, and this quantile is taken as the value of p.

However, when testing the goodness-of-fit of a distribution fit from a parametrized family of distributions, the Kolmogorov distribution is no longer the appropriate target of comparison for D_{KS}. Consequently, we employee a method known as *parametric bootstrapping* [Clauset *et al.*, 2009]. In parametric bootstrapping, the p value is based not only on the test statistic (e.g., in our case D_{KS}) computed over the observed data and the fitted hypothesized distribution, but also over synthetic data drawn from this fitted distribution and refits of the hypothesis model to these synthetic data.

Specially, we first fit our model to the observed data by estimating the parameters using Maximum Likelihood Estimation (MLE). MLE chooses the parameters which maximize the likelihood of observing the data under the model. Then we calculate the test statistic over the observed data and our fitted model. Next, we draw b synthetic data sets from the fitted model using the parameters we have just estimated. For each, we run MLE again to estimate the parameters of a new model from the hypothesized family of distributions, and then calculate the test statistic between the synthetic sample and its specially estimated private model. The p-value is then defined to be the fraction of the synthetic data whose test value is larger than the test value obtained from observed data. Algorithm 1 shows the steps of this parametric bootstrapping algorithm formally. The effect is a Kolmogorov-Smirnov (KS) test that is unbiased when used on hypotheses involving fitted families of distributions. The larger the number of synthetic data sets generated, b, the more reliable the p-value obtained. In our analysis we let $b = 1000$.

Our data was processed in Matlab, and analyzed in R. We used the *copula* [Yan, 2007] and *CDVine* [Brechmann and Schepsmeier, 2013] packages to perform our copula-based fits and as the basis for implementing Algorithm 1.

5 Results

Using this framework, we next fit models of CPU and memory usage both separately, and subsequently together as a joint model of demand.

5.1 Fitting the Marginals

In both the CPU and memory usage data, there are generally sufficiently large numbers of big jobs that we expect heavy-
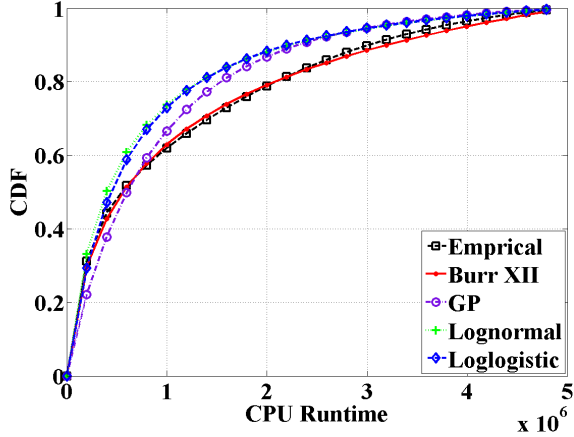
Figure 1: The empirical CDF of CPU usage, and MLE for the Burr XII, Generalized Pareto, Log-Normal and Log-Logistic distributions.
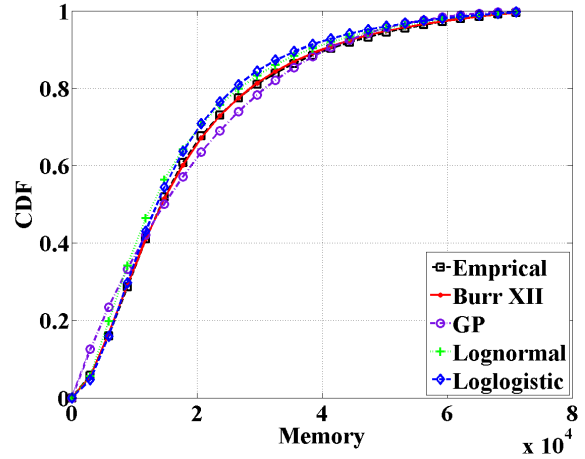

Figure 2: The empirical CDF of memory usage and MLE fits of the Burr XII, Generalized Pareto, Log-Normal and Log-Logistic distributions.

tailed distributions to perform well. We fit our data using Burr XII, Log-Normal, Generalized Pareto and Log-Logistic distributions by MLE. Table 2 lists the parameters as computed.[1]

To evaluate the model fit, we begin by visually observing the correspondence between the empirical CDF and the CDF of each distribution in Table 2. Figures 1 and 2 illustrate this for CPU and memory usage respectively. We observe that the truncated Burr XII distribution is the closest to the empirical distribution for both CPU runtime and memory usage.

To back up the intuition we obtain from the graphs, we formally evaluate the goodness-of-fit of each distribution using the bootstrap KS method from Section 4, evaluating the CPU and memory data separately. Accordingly, we sample 5000 data points and run our KS test procedure using the distributions and parameters from Table 2 as our hypothesis. The resulting p-value is also shown in Table 2. From the table it is clear that Log-Normal, Log-Logistic and Pareto distribution perform poorly, with the differences between their fitted distributions and the data being strongly statistically significant. The calculated p-value of the Burr XII is very large, indicating no evidence to distinguish it from the observed data for CPU and memory. We therefore adopt it as our marginal model of both resources (with different parameters for each).

5.2 Fitting the Copula

Having created a model for each of the marginal distributions, we next turn to modeling the copula needed to relate them. To get a sense of the joint data (and to use as a baseline comparison for the fidelity of our eventual model) we show a scatter/histogram plot of a 2500 point sample of the data in Figure 3.

Next, in Figure 4 we apply the probability integral transform described in Section 2.3 to the data in Figure 3. Specifically, we use the Burr XII marginal distributions for both CPU and memory usage with the parameters provided in Ta-

---

[1]We fit truncated distributions, at $5 * 10^6$ for CPU and $7.5 * 10^4$ for memory.

Distribution	Estimated Parameter	Copula p-value	End-to-End p-value
Gaussian	0.329	0.08	0.15
Frank	1.993	0.26	0.44
Gumbel	1.226	0.03	0.08
T Copula	0.272	0.005	0.05
Clayton	0.329	0.17	0.41

Table 3: For each copula we show its MLE parameters and its p-value as calculated by our parametric bootstrap KS test on the copula alone. Lastly, we provide the p-value of a two-sample bootstrap KS-test of the full joint model to the observed data.

ble 2. The result is the empirical copula, i.e. the "coupling" structure manifest in the data once the marginal distributions have been made uniform. We can see that our truncated Burr XII marginal distribution is doing an effective job at capturing most of the marginal structure, and thus the probability integral transform yields data that is nearly uniform in each dimension.

Having applied our marginal models, we next seek a parametric copula that captures the structure observed in Figure 4. Accordingly, we fit five of the most commonly used families of copula including Gaussian, Frank, Gumbel, T, and Clayton. Table 3 shows the estimated parameters for each family. To get a sense of the copulae produced, we plot a random sample from the fitted Frank copula in Figure 5. We can see that this plot is a close match to the transformed empirical data shown in Figure 4. The Frank copula, as observed in Figures 5 has weak correlation in the tails compared to other copula [Trivedi and Zimmer, 2007]. However, there is still a tendency for there to be stronger correlation between CPU and Memory among either extremely large or extremely small jobs, which is reflected in our data set.

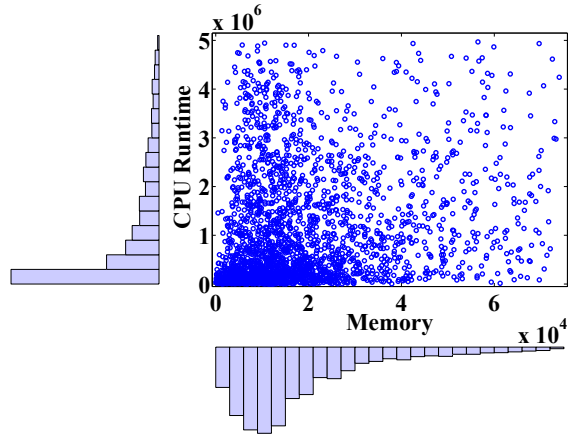Next, to evaluate goodness-of-fit, we apply our bootstrap

Figure 3: A 2500 point sample of the observed Google CPU and memory usage data.
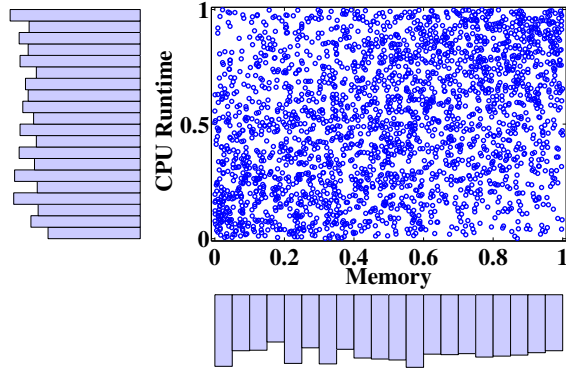

Figure 5: A draw of 2500 random points from the Frank copula using the parameters in Table 3.
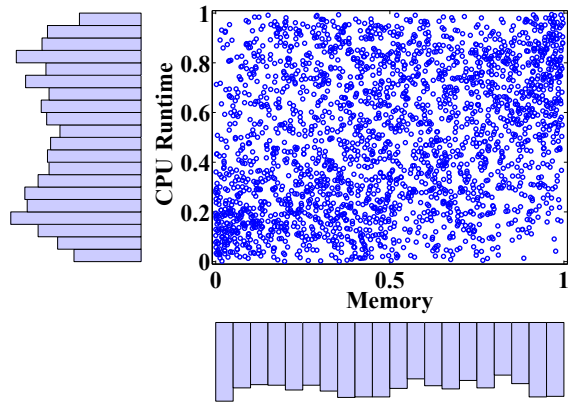

Figure 4: The probability integral transform applied to the usage data in figure 3, based on a Burr XII fit according to the values provided in Table 2. This is an empirical copula when the margins have been described by our Burr XII model.
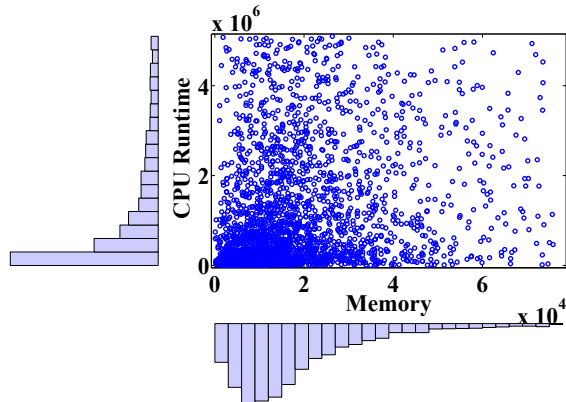

Figure 6: A draw of 2500 points from our full end-to-end multivariate model, using the Frank copula parametrized in Table 3 and illustrated in Figure 5. Clearly the model closely matches the original data from Figure 3.

KS test algorithm, but now on the bivariate copulae instead of univariate marginals. The resulting p-values are listed in Table 3. From the table it is clear that the p-value for all the copulae except Frank and Clayton copula are small, indicating evidence to reject them. These large values for the Frank and Clayton indicate that we have no evidence to reject them as reasonable models of the coupling between the CPU and memory properties. Although both the Frank and the Clayton are reasonable models, the evidence for the Frank is stronger than the Clayton.

The last step of validating our model is to evaluate the complete model including both the fitted Burr XII marginals and the Frank copula together. Figure 6 shows a random sample of 2500 pints from this model, and appears nearly identical to the original data in Figure 3. Once again, we can formalize this by using a KS test to compare the observed data and data drawn from our fitted model. Table 3 shows the resulting

p-values, for each copula we considered. The large p-value shown for the Frank copula indicates that our aggregate Burr XII-copula model can not be statistically differentiated from the data, and thus forms a valid model of joint demand.

6 Conclusion

In this paper, we have analyzed the CPU and memory usage of jobs separately and as a joint distribution using real data from Google data-centers. Our marginal models of individual process attributes (like CPU and memory) have better fidelity than those used previously. We obtained this improvement by using a more flexible distribution, the Burr XII, than previously considered. We then leveraged this accuracy in modeling marginal distributions to create the first effective multi-attribute model of data-center demand. Concretely, we show that an Archimedean copula, specifically the Frank, can be used to great effect in modeling the "coupling" structure

between the attributes of computational processes in distribution. Then, by putting both our Burr XII marginal models and our Frank copula model together, we obtain a highly accurate model of the full joint distribution across attributes.

Understanding multivariate resource usage distributions is critical to creating better resource allocation and scheduling algorithms for large computing centers, which in turn offer the promise of significant energy savings and CO_2 reduction.

References

[Agrawala *et al.*, 1976] Ashok K Agrawala, JM Mohr, and RM Bryant. An approach to the workload characterization problem. *Computer*, 9(6):18–32, 1976.

[Beloglazov and Buyya, 2010] A. Beloglazov and R. Buyya. Energy efficient allocation of virtual machines in cloud data centers. In *Proceedings of 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid 2010)*, pages 577–578, May 2010.

[Beloglazov *et al.*, 2012] Anton Beloglazov, Jemal Abawajy, and Rajkumar Buyya. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future Generation Computer Systems*, 28(5):755–768, 2012. Special Section: Energy efficiency in large-scale distributed systems.

[Brechmann and Schepsmeier, 2013] Eike Christian Brechmann and Ulf Schepsmeier. Modeling dependence with C-and D-vine copulas: The R-package CDVine. *Journal of Statistical Software*, 52(3):1–27, 2013.

[Burr, 1942] Irving W. Burr. Cumulative frequency functions. *The Annals of Mathematical Statistics*, 13(2):215–232, 1942.

[Burr, 1968] Irving W. Burr. On a general system of distributions: Iii. the sample range. *Journal of the American Statistical Association*, 63(322):636–643, 1968.

[Chiang and Vernon, 2001] Su-Hui Chiang and Mary K Vernon. Characteristics of a large shared memory production workload. In *Proceedings of the 7th International Workshop on Job Scheduling Strategies for Parallel Processing (JSSPP-01)*, pages 159–187, June 2001.

[Clauset *et al.*, 2009] A. Clauset, C. Shalizi, and M. Newman. Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703, 2009.

[D'Agostino, 1986] Ralph B D'Agostino. *Goodness-of-Fit Techniques*, volume 68. CRC press, 1986.

[EIA, 2015] U.S. energy information administration frequently asked questions. U.S. EIA, 2015. `http://www.eia.gov/tools/faqs/faq.cfm?id=97&t=3`.

[Guevara *et al.*, 2014] Marisabel Guevara, Benjamin Lubin, and Benjamin C. Lee. Market mechanisms for managing datacenters with heterogeneous microarchitectures. *ACM Transactions on Compututer Systems*, 32(1):3:1–3:31, Feb 2014.

[Harchol-Balter and Downey, 1997] Mor   Harchol-Balter and Allen B. Downey. Exploiting process lifetime distributions for dynamic load balancing. *ACM Transactions Computer Systems*, 15(3):253–285, Aug 1997.

[Leland and Ott, 1986] Will Leland and Teunis J. Ott. Load-balancing heuristics and process behavior. *SIGMETRICS Performance Evaluation Revue*, 14(1):54–69, May 1986.

[Li *et al.*, 2004] Hui Li, David Groep, and Lex Wolters. Workload characteristics of a multi-cluster supercomputer. In *Proceedings Of the 10th International Workshop on Job Scheduling Strategies for Parallel Processing (JSSPP-04)*, pages 176–193, June 2004.

[Lubin *et al.*, 2009] Benjamin Lubin, Jeffrey O Kephart, Rajarshi Das, and David C Parkes. Expressive power-based resource allocation for data centers. In *Proceedings Of the 21st International Joint Conference on Artificial Intelligence (IJCAI-09)*, pages 1451–1456, 2009.

[NRDC, 2013] America's data center efficiency assessment. National Resources Defense Council, 2013. `http://www.nrdc.org/energy/data-center-efficiency-assessment.asp`.

[Parsons and Sevcik, 1996] Eric W. Parsons and Kenneth C. Sevcik. Coordinated allocation of memory and processors in multiprocessors. *SIGMETRICS Performance Evaluation Revue*, 24(1):57–67, May 1996.

[Reiss *et al.*, 2011] Charles Reiss, John Wilkes, and Joseph L Hellerstein. Google cluster-usage traces: Format + schema. *Google Inc., White Paper*, 2011.

[Rosin, 1965] Robert F. Rosin. Determining a computing center environment. *Communications of the ACM*, 8(7):463–468, Jul 1965.

[Sklar, 1959] A. Sklar. *Fonctions de répartition à n dimensions et leurs marges*, volume 8. Publications de l'Institut de Statistique de L'Université de Paris, 1959.

[Tadikamalla, 1980] Pandu R. Tadikamalla. A look at the burr and related distributions. *International Statistical Review*, 48(3):337–344, 1980.

[Trivedi and Zimmer, 2007] Pravin K Trivedi and David M Zimmer. *Copula Modeling: an Introduction for Practitioners*. Now Publishers Inc, 2007.

[Walter and Wallace, 1967] Edward S. Walter and Victor L. Wallace. Further analysis of a computing center environment. *Communications of the ACM*, 10(5):266–272, May 1967.

[Yan, 2007] Jun Yan. Enjoy the joy of copulas: with a package copula. *Journal of Statistical Software*, 21(4):1–21, 2007.