Prediction and Control of Bursty Cloud Workloads: A Fractal Framework

Mahboobeh Ghorbani, Yanzhi Wang, Massoud Pedram and Paul Bogdan
Electrical Engineering Department, University of Southern California, Los Angeles, CA
mahboobg,yanzhiwa,yuankunx,pedram,pbogdan @usc.edu

ABSTRACT

Cloud Computing is a promising approach to handle the growing needs for computation and storage in an efficient and cost-effective manner. Towards this end, characterizing workloads in the cloud infrastructure (e.g., a data center) is essential for performing cloud optimizations such as resource provisioning and energy minimization. However, there is a huge gap between the characteristics of actual workloads (e.g., they tend to be bursty and exhibit fractal behavior) and existing cloud optimization algorithms, which tend to rely on simplistic assumptions about the workloads. To close this gap, based on fractional calculus concepts, we present a fractal model to account for the complex dynamics of cloud computing workloads (i.e., the number of request arrivals or CPU/memory usage during each time interval). More precisely, we introduce a fractal operator to account for the time-varying fractal properties of the cloud workloads. In addition, we present an efficient (online) parameter estimation algorithm, an accurate forecasting strategy, and a novel fractal-based model predictive control approach for optimizing the CPU utilization, and hence, the overall energy consumption in the system while satisfying networked architecture performance constraints like queue capacities. We demonstrate advantages of our fractal model in forecasting the complex cloud computing dynamics over conventional (non-fractal) models by using real-world cloud (Google) traces. Unlike non-fractal models, which have very poor prediction capabilities under bursty workload conditions, our fractal model can accurately predict bursty request processes, which is crucial for cloud computing workload forecasting. Finally, experimental results demonstrate that the fractal model based optimization outperforms the non-fractal based ones in terms of minimizing the resource utilization by an average of 30%.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

General Terms

Theory

1. INTRODUCTION AND CONTRIBUTION

Cloud computing, which shifts the computation and storage resources from network edges to a "Cloud" from which businesses and users are able to access vast resources on demand from anywhere in the world [1, 2, 10], is growing rapidly and becoming widely adopted by individuals, corporations, businesses, and governments. Major cloud service providers, including Google, Microsoft, Amazon, etc., have built and continue to build large data centers with massive computation and storage capabilities and charge clients for accessing resources or services provided by these data centers. The energy consumption of data centers is rapidly increasing and covers up to 2 percent of the total electrical energy consumption in the U.S. in 2010 [5]. For example, Microsoft's data center in Quincy, Washington consumes 48 megawatts which is enough to power 40,000 U.S. homes [4]. This fact motivates the need to develop resource management strategies for the cloud infrastructure in order to reduce their energy consumption and total cost of ownership.

The problem of resource provisioning in the Cloud Computing infrastructure has been addressed extensively in the previous work [8, 11]. Typically the goal is to minimize the total energy consumption of the data center while satisfying response time constraints specified in Service Level Agreements (SLAs). Different optimization approaches based on queueing theory, (stochastic) binpacking heuristics, or network flow theory have been presented, each with some advantages and disadvantages compared to the others. However, an important shortcoming of these approaches is their reliance on simplistic models of the cloud workload. In particular, key assumptions behind all of these approaches are as follows: (1) cloud computing workloads, i.e., the numbers of service request arrivals or CPU/memoery usage in different time intervals, are memoryless, stationary, and/or periodic and (2) cloud workloads can be accurately estimated and predicted by non-fractal models, i.e., cloud workloads exhibit no long-range dependency. However, as we will show in Section 3 and 4, such traditional and non-fractal approaches fail to fully capture key characteristics of cloud computing workloads, namely, the long-range memory dynamics and the highly varying nature of cloud workloads (see Figure 6). In reality, ignoring the long-range correlations and relying on prediction results of non-fractal models can lead to very poor (or even embarrassing) workload fore-

casting and resource provisioning/control results.

To overcome this important shortcoming and address these challenges with respect to resource provisioning in the cloud infrastructure, we make the following novel contributions in this work:

- We present a fractal model for capturing key characteristics of cloud computing workloads such as the non-stationary and long-range dependence properties. Our proposed model not only improves the goodness-of-fit figure-of-merit when compared to non-fractal approaches, but also offers significantly better prediction capabilities of future cloud workloads (see Figure 4 to Figure 6). In addition, our newly proposed model reduces the number of parameters required in the non-fractal models for dynamically capturing the historical trends of a cloud workload to a single fractal exponent.
- By accurately capturing the complex dynamics of cloud workloads, our model not only leads to better prediction results, but also results in better utilization and provisioning results of cloud resources, and thus, significant energy savings (see Section 4). Simply speaking, our fractal model with much fewer parameters, but with smart accounting of nonlinear dynamics of cloud workloads, helps maximize energy efficiency of cloud infrastructures.
- To verify the accuracy of the proposed model and demonstrate the benefits of our fractal optimal control strategy, we develop a simulation environment that takes as input real-world workloads such as Google traces [15].
- To prove the feasibility of the proposed fractal optimal control algorithm, we develop a wavelet-based technique for online estimation of the fractal model parameters (cf. Equations (1) and (3)) and reduce the optimality conditions to solving a sparse linear programming problem. To ensure that the execution of prediction algorithm can be achieved in real-time, we provide a hardware implementation.

The rest of the paper is organized as follows. In Section 2 we discuss the related work and motivation. Section 3 explains the fractal-based modeling and optimization in the context of cloud computing whereas Section 4 presents our methodology and experimental results. Conclusion is provided in Section 5.

2. RELATED WORK AND MOTIVATION

2.1 Related Work

Establishing itself as a new computing paradigm, cloud computing aims to sustain the execution of a large heterogeneous set of applications on distributed computational modules (e.g., servers, server clusters, and data centers) that efficiently communicate with each other through a real-time communication network (i.e., Internet). Since its first definition in 1995 by Compaq researchers, numerous research efforts have tackled various cloud computing optimizations: task placement [11], load balancing and scheduling [6, 22], resource allocation/consolidation [8, 9, 17, 19, 16, 23, 25, 26, 28, 31, 32]. However, neglecting the nature of incoming workload results in poor provisioning, causes excessive resource utilization, high power consumption and/or service level agreement (SLA) violations. Consequently, concerns regarding the energy consumption, cooling and carbon emission costs have led to active research on dynamic optimization for cloud computing platforms. Relying on queuing theory, convex optimization, and/or control theory, several approaches [12, 14, 22, 27, 29, 33] seek to determine the best allocation of computing and/or storage resources such that user demands are satisfied. For instance, Gandhi et al. [14] proposed a technique which uses a predictive algorithm to allocate resources at coarse time scales and a reactive controller to mitigate inaccuracies in previous decisions over shorter time scales. It assumes that the prediction errors are small and that workloads are periodic and slowly-varying. Along the same lines, Yao et al. [33] used an auto-regressive moving average (ARMA) model for incoming workload prediction and developed a model predictive control approach for minimizing the electricity cost subject to power budget constraints. Major assumptions behind the above-mentioned approaches are as follows: cloud computing workloads are (1) memoryless, (2) stationary, or (3) periodic and can be well fitted and predicted by non-fractal models. However, the actual cloud computing workloads are highly time-varying in nature.

There are also research work that analyze characteristics of cloud computing workloads with the conclusion that such workloads are not stationary. For example, [38] discusses the heterogeneity and dynamicity of cloud workloads and denies the usage of some popular simplified assumptions including Poisson arrival rate and Gaussian distribution for the task duration. [39] compares the workload in cloud computing versus grid computing and identifies a number of differences between the two in terms of job/task length, job priority, machine utilization level, etc. [40] focuses on the frequency and pattern of machine maintenance events, job and task level workload behavior, and how the overall resource on a server cluster is used. In this paper, besides the heterogeneity and dynamicity, we confirm the fractality of workloads by using the Detrend Fluctuation Analysis (DFA) [7], and propose a forecasting and control algorithm on the basis of the time-dependent fractal model. We demonstrate superiority of our approach compared to non-fractal models by extensive experiments on real cloud workloads from Google Production Center [15].

There are several works addressing the observed self-similarity in internet traffic [46][47]. However, they are for micro-scale observation and they are not for this macro-scale cloud computing workloads submitted to a data center. On the other hand, there is a research sturdy by [45] that reports observed self-similarity and fractality in power consumption traces of Microsoft data center. However, they do not use fractional calculus as a tool for prediction and control of data center management as we propose in this work. Also, there are some works which address similar problem of this paper (prediction and control of resource utilizations in cloud computing workloads) [48]. However, they have not count for self-similarity and fractality of the workload and they have modeled workload as a memoryless autoregression moving average models (ARIMA). Here, we compare our results with the ones that can be obtained from these memoryless models.

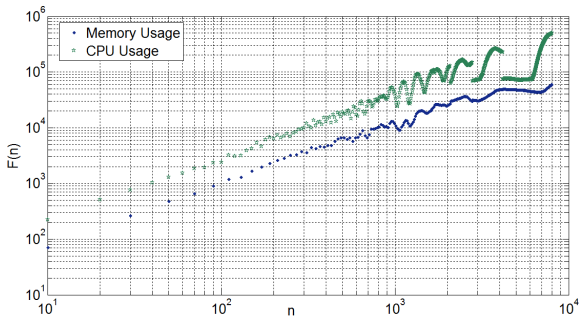

Figure 1: Detrended Fluctuation Analysis for CPU and Memory usage in Google Cloud Computing Workloads

2.2 Motivation

The state-of-the-art mathematical modeling and optimization algorithms in cloud computing mainly rely on queuing theory and the assumptions that memoryless property and Gaussian distribution hold for the cloud computing workloads (the workload refers to the number of request arrivals, or CPU/memory usage in each time interval). However, as observed from real world traces like the Google's Production Center trace [15], conventional models cannot offer a good estimate of the real data. We investigated the fractal nature of the cloud computing workload by applying the Detrended Fluctuation Analysis (DFA) method [7] on the CPU and memory demand of all jobs run at the Google's Production Center trace [15]. The DFA method shows that the Hurst exponents for CPU and memory usage trace of the system are 0.89 and 0.84, respectively, which proves the long-range dependency (normally long-range dependency property is prominent when the Hurst exponent is larger than 0.5.) Results are presented in Figure 1 where green line shows DFA for the CPU usage in the system and blue line represents the DFA analysis for the memory usage of all jobs in the data center. This analysis proves that the extracted workload from Google traces exhibits long-range dependence.

One of the main concerns in today's datacenter provisioning is the power efficiency. Histogram of average CPU utilization for more than 5000 Google's servers and the energy efficiency as a function of CPU utilization for severs is presented in [34]. They show that at idle state when no useful job is performed (and the utilization is zero) about 50% percent of peak power at highest utilization is consumed. Also, servers usually operate as utilization levels with low energy efficiency as between 20% to 60%. Another results gathered from [15] in Figure 2 also show the difference between CPU and memory usage of the tasks with real allocated CPU and memory resources in the data center. These two observations verify each other in the sense that when resource (e.g. CPU or memory) allocation is less than the actual usage, most of the allocated resources perform in low utilization levels. On the other hand, on the basis of energy efficient plot in [34], operation at low utilization levels contradicts energy efficiency. For instance. two servers performing at low utilization levels like in 35% consume 25% more energy than a single server performing at 70% utilization level. On the basis of the observed deficiency in provisioning cloud

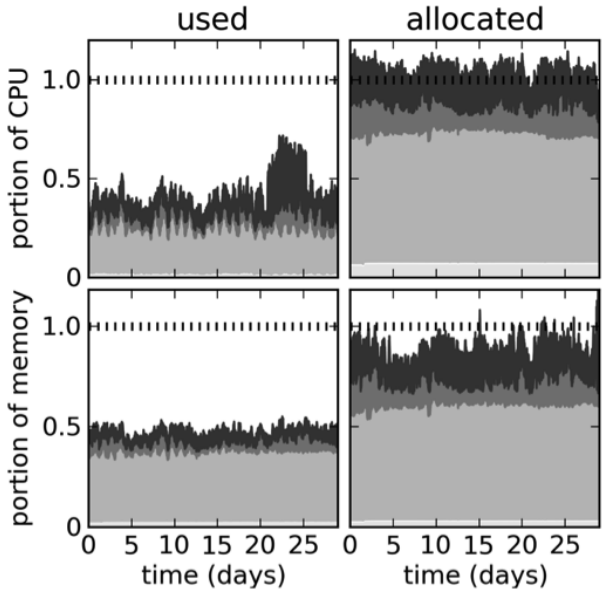

Figure 2: CPU and memory usage of google workloads in a Google's 12k-machine cell over about a month-long period [38]

workloads, we propose a fractal model for prediction and optimization. In other words, we aim to provision the utilization of datacenter workloads to shift the distribution of utilization level to more energy efficient values as much as possible.

Fractal optimal control approach has been used in other engineering applications. Fo instance, a fractional state equation was proposed in [44] for modeling the core and uncore workloads. This fractional calculus based modeling enabled the development of a power use fractional optimal control approach for power management in network on chip (Noc) [44]. They propose power optimization based on fractal state equations for voltage and frequency scaling. Also the work in [43] use fractal optimal control approach for insulin regulation in artificial pancreas. Their contribution relies on modeling time-dependent fractal order of blood glucose time series. However, none of these works include prediction algorithm. Morover, the problem definition and optimization framework are different.

3. FRACTAL APPROACH TO CLOUD COMPUTING OPTIMIZATION

On the basis of the observed non-stationary nature of cloud workloads, we use a two-phase resource utilization provisioning, called 'predictive and reactive provisioning' as depicted in Figure 3. First the forecasting module predicts the workload for the next control horizon, and then the controller estimates the number of necessary resources (e.g., processing cores) for the predictable part of incoming load. Since any forecasting algorithm suffers from non-avoidable errors, the system allocates extra resources that can be used to serve the unpredicted load, called spare pool in Figure 3. This allocation is made based on the history of the system operation. In the following two subsections, we explain the workload modeling and resource utilization optimization.

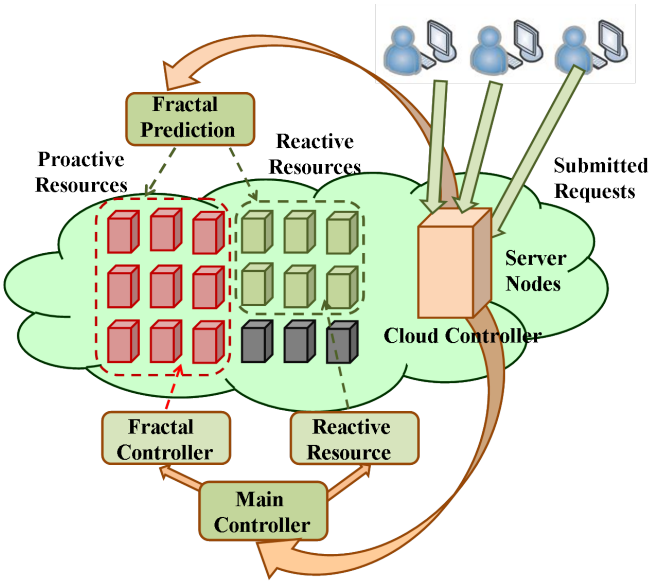

Figure 3: Predictive and Reactive System Provisioning

3.1 Fractal Modeling of Cloud Workloads

Inspired by the observation about the non-stationary and fractal nature of request arrivals in cloud computing infrastructure, we model the number of arrival requests ($X(t)$) in each time interval using the following time-dependent fractional order equation:

$$\frac{d^{\alpha}X(t)}{dt^{\alpha}} = a(t)X(t) + b(t) \tag{1}$$

In this model, $a(t)$ and $b(t)$ are linear regression model parameters and α is the order of the fractional derivative, which captures the long-range dependence of the system and has the following definition:

$$\frac{d^{\alpha}X(t)}{dt^{\alpha}} = \lim_{\Delta t \to 0} \frac{1}{\Delta t^{\alpha}} \sum_{j=0}^{[t/\Delta t]} (-1)^{j} \cdot \binom{\alpha}{j} \cdot X(t - j\Delta t) \tag{2}$$

Equation (2) describes how the current value of the workload (e.g. number of incoming requests in some fixed time interval) depends on the previous values of the workload. Extracting the mathematical models of the dynamics of time series has the following implications:

- Extracting a good model for describing the dynamics of the system enables one to establish a method for forecasting. Forecasting results are essential for making decisions about performance of the system. In cloud computing framework, for example, being able to predict the number of arrival requests in the system in specific time intervals plays an important role in determining decision variables like the utilization degree of the system. Poor prediction leads to poor utilization of the system resources e.g. high energy consumption or reduced quality of service.
- The mathematical model is used to formulate the dynamical equation of the system which determines the relation between changes in the workload and utilization of the data center servers. Since any optimization problem relies on modeling the dynamical equations in the system, proposing accurate mathematical model enables the optimization engine to rely on a more realistic model, which determines the quality and performance of the control algorithm. For example, knowing that the number of waiting requests in the system follows a fractal model (as shown in Figure 1 and the discussions above) enables us to write a dynamical equation (Equation (3)) which uses a fractal order derivative instead of integer order derivative (e.g. first order derivative).

A comparison of fractal and non-fractal models is presented in Section 4 and the benefits of the proposed fractal model for highly fluctuating workloads are verified by extensive experiments from Google traces [15].

3.2 Fractal Optimal Control Formulation

Given scarce resources and tight energy budget (to limit the electrical energy bill of the owner) in the cloud computing infrastructure, the goal of the optimization algorithm is to schedule and dispatch the coming requests and assign utilization level of the servers to minimize energy consumption and satisfy the performance requirements.

To formulate the problem we assume that there are N servers each having maximum of M CPU resources to deliver service to the incoming requests. Also we have assumed that there is an optimization horizon of length H seconds which can vary according to available technology limits for latency of changing voltage and frequency and also delay turning processors On and Off. At the beginning of each horizon, the optimization algorithm decides about (a) the probability of dispatching incoming requests during that horizon to each server ($\mu_j(t)$) and (b) the utilization level of the servers ($u_j(t)$). Motivated by the analysis of the Google traces and the observation of long-range dependent nature for the number of remaining requests in the system ($R_j(t)$), we advocate using a fractal dynamical equation for describing the system state for each j^{th} server:

$$\frac{d^{\alpha}R_j(t)}{dt^{\alpha}} = a(t)X_j(t) + b(t)u_j(t) \tag{3}$$

Where the remaining number of requests ($R_j(t)$) defines the system state and $X_j(t) = \mu_j X(t)$ denotes the portion of all incoming requests dispatched to j^{th} server. Since not all the requests during each time horizon come at the beginning, the role of the prediction algorithm is crucial to estimate $X(t)$. Consequently, the goal of the optimization problem is to minimize:

$$Energy = \sum_{t=0}^{H} \sum_{j=1}^{N} Energy_{j,t}(u_j(t)) \tag{4}$$

where $Energy_{j,t}(u_j(t))$ is the energy consumption of server j at utilization level $u_j(t)$. This problem is subject to the following constraints:

$$\sum_{i=1}^{N} \mu_j = 1 \quad (5)$$

$$u_{min} \leq u_j(t) \leq u_{max}, \text{ for } j = 1 \text{ to } N \quad (6)$$

$$0 \leq R_j(t) \leq R_{max}, \text{ for } j = 1 \text{ to } N \quad (7)$$

$$\sum_{j=1}^{N} R_j(H) \leq R_{reference} \quad (8)$$

The first constraint points to the fact that the sum of all probability variables should be one. The second set of constraints bound the utilization level of the servers to the desired level which is typically set to a value close to but less than one to avoid excessive delay penalty associated with long waiting times for the requests when the resources are fully utilized. The third set of constraints point to the queuing capacity limitation of the system and can also be interpreted as the response time limit. The last constraints limit the remaining pending request at the end of control horizon in order to guarantee performance constraints. For this formulation we have assumed that power consumption of each sever has a linear relations to the CPU utilization level ($u_j(t)$) according the real measurement from Google servers reported by [35].

The above formulation relies on knowing the number of requests arriving in the system in each time interval. However, not all the requests during each time interval come at the beginning of the interval as we observed in Google traces [15]. So, the optimization algorithm needs to forecast $X(t)$ in the beginning of each time interval in order to take appropriate decisions. We have shown forecasted value of $X(t)$ by $\hat{X}(t)$. The entire optimization strategy is verified and the results are presented in Section 4.

4. METHODOLOGY AND EXPERIMENTAL RESULTS

We have conducted our analysis and experiments on workload traces for a production compute cluster at Google [15] consisting of approximately 12K machines. The dataset was released on November 29, 2011. The workload traces contain scheduling events as well as resource demand and usage records for a total of 672003 jobs and 25462157 tasks over a time span of 29 days. Specifically, a job is an application that consists of one or more tasks. Each task is scheduled on a single physical machine. When a job is submitted, the user can specify the maximum allowed resource demand for each task in terms of required CPU, memory and disk size. At run time, the usage of a task measures the actual consumption of each type of resources. The current Google cluster traces provide task demand and usage for CPU, memory and disk. The usage of each type of resource is reported at 5 minute intervals. We mainly focus on CPU and memory, as they are typically scarce compared to disk. However, we believe it is straightforward to extend our approach to consider other resources such as disk space.

We applied our prediction experiments on the CPU and memory usage time series in Section 4.1 and test the performance of the fractal control algorithm in Section 4.2.

4.1 Model Identification and Forecasting

Motivated by the non-stationary and fractal nature of cloud workloads, we develop an efficient wavelet-based estimation

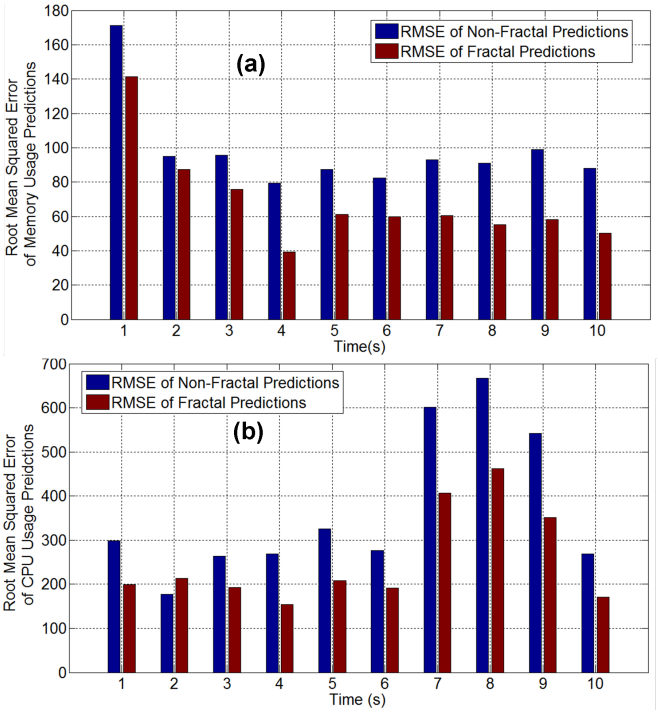

Figure 4: RMSE of predictions results of fractal vs. non-fractal model for CPU and Memory usage traces

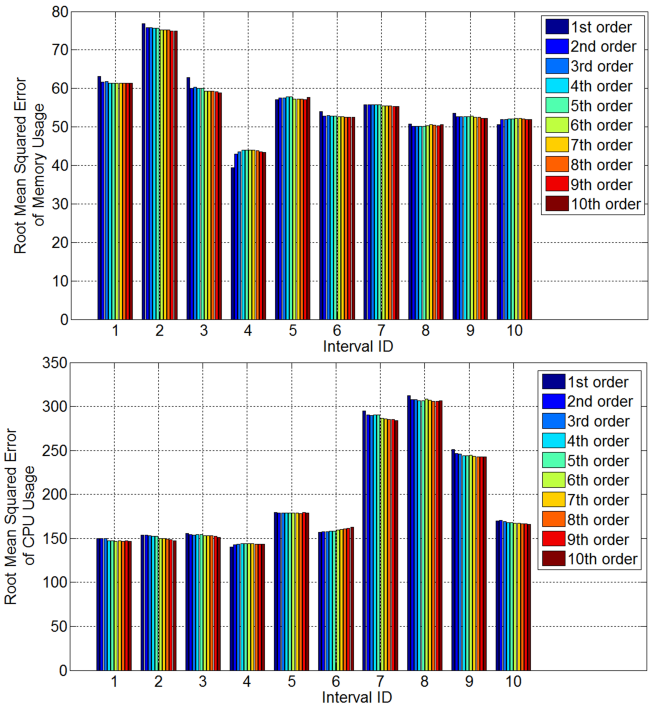

Figure 5: RMSE of prediction results of non-fractal model by increasing the order

method for fractional order derivative parameter α (see Equation (6)). Wavelets [30] provide a strong tool to capture both short-lived time phenomena like singularity points and also identify the fractal (scaling) behavior in stochastic processes because of localization in different scales. To sum up, the model identification consists of two steps: (i) a wavelet transform which estimates the order exponent α; and (ii) a low dimensional linear regression model to determine the remaining model parameters from Equation (6).

Building on our findings, we develop a wavelet-based forecasting algorithm which proceeds as follows:

1. The algorithm first uses detrended fluctuation analysis [42] and uses a linear regression model to estimate the fractional order derivative α near the forecasting point.

2. Upon estimating the fractional order derivate (α), we obtain $W(t)$, which is the smooth component of the workload by applying the fractional order derivative:

$$\frac{d^{\alpha}X(t)}{dt^{\alpha}} = W(t) \tag{9}$$

3. The algorithm then fits an ordinary linear regression model to the $W(t)$ time series till the desired forecasting point, then it generates the forecasted $\hat{W}(t)$ time series:

$$W(t+1) = aW(t) + b \tag{10}$$

4. Extracts $\hat{X}(t)$ which is the forecasting result for the original time series using Equation (2). That is, we can finally obtain $\hat{X}(t)$ by using Equation (6) given $W(t)$.

5. Finish

We examine the above algorithm on the real Google traces extracted from [15]. We compare the results of fractal forecasting algorithm to the non-fractal model of 10-order back shift operator in terms of ratio of Root Mean Square Error (RMSE) to the real observations in the periods of 8 hours with one minute granularity. We used the algorithm for both CPU and Memory usage time series. Of note, our fractal model consists of α, a and b and hence, significantly fewer parameters. Figure 4 shows that the proposed fractal model has smaller RMSE than non-fractal one for various cloud computing traces. This also proves that our fractal model has better accuracy than the non-fractal one. We also examine the amount of improvement of prediction results of non-fractal model by increasing number of parameters. As it is shown in Figure 5, increasing order of the non-fractal model modestly improves the accuracy of the model and it is not comparable to fractal model with only one back shift operator.

Since the average is not the only metric for assessing the prediction of fractal model, we show the predication results of both fractal and non-fractal models over one sample 12-hour period. The superiority of the fractal model in capturing spikes is the most interesting result we found. This is very critical in cloud computing infrastructure since it requires *to maintain quality of service under bursty conditions*. Also, we depict one sample forecasting result of fractal and non-fractal model in Figure 6. In Figure 6, part (a) and (b) show the prediction results in a 12-hour period for both methods and part (c) and (d) show the results of both methods in a more zoomed picture where the superiority of the fractal method in capturing spikes is more obvious.

4.2 Performance of fractal controller

We use power model proposed in [35] for estimating total power consumption as a function of CPU utilization. According to this model, CPU utilization signal alone is a good estimate of the total power consumption in data center on the basis of the measured data.

Motivated by the idea of using forecasting for resource utilization in Section 3, we use a two-phase resource provisioning named predictive and reactive provisioning as depicted in Figure 7. Although fractal model has better prediction capabilities compared to non-fractal one, any forecasting algorithm may have non-avoidable error. Consequently, any dynamic optimization should provide enough amount of resources ready to compensate for unpredicted load. As depicted in Figure 7, at the beginning of each time frame (interval), the controller estimates the number of resources necessary for the predictable part of incoming load as formulated in Section 3.2 via an optimal control algorithm, and also reserves a spare resource pool ready for reactive load which is the error of the prediction. To compare the impact of using a fractal vs. non-fractal model for optimizing the resource utilization, we report the results in two phases:

Savings due to fractal controller: To study the impact of the performance of the fractional order controller discussed in Section 3, we consider several time intervals from the Google traces [15], use the forecasting algorithm discussed in Section 4.1, and solve the optimization problem outlined in Section 3.2. To compare our approach with traditional memoryless approaches, we also solve the non-fractal optimal control problem for which the forecasting is based on integer order dynamical equation. We apply both controllers and extracted the decision variables and show how the fractal order controller outperforms the integer order controller in terms of minimizing the resource utilization while satisfying systems constraints. Part (a) of Figure 7 shows the energy savings due to using fractal controller over 10 control horizons.

Savings due to fractal prediction: Our resource utilization provisioning module estimates the worst case of prediction error (based on the previous observation) and allocates a spare pool of resources for unpredicted load. If some prediction error happens beyond that estimation, it would be subject to delay in getting the resource till the system allocates available resources. For the sake of comparison, we assume the same scheme for non-fractal model and show the resource savings of using fractal vs. non-fractal model in part (b) of Figure 7 over 10 control horizons.

Overall Savings: By adding the savings of both reactive and predictive part, we show in part (c) of Figure 7 the energy savings due to using fractal model as a whole. Though the prediction results of fractal model have significant improvement, the reactive portion of the resource are not very

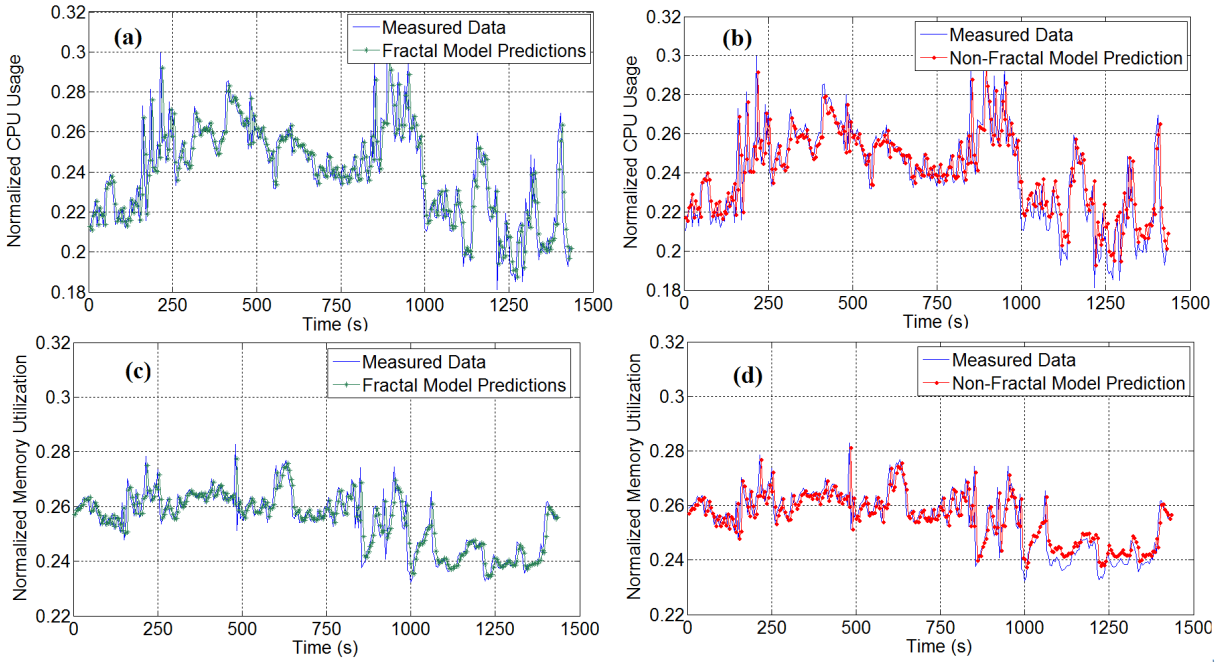

Figure 6: Fractal and Non-fractal prediction results (a,b). Prediction results of CPU usage in Google traces [15] by fractal approach is shown by green in part (a) and non-fractal one is shown by red in part (b) in comparison to the measured values in blue, (c,d). Prediction results of Memory usage in Google traces [15] by fractal approach is shown by green in part (c) and non-fractal approach is shown by red in part (d) in comparison to the measured values in blue

dominant. As a result, most of the gain obtained by the fractal model is about the same range of gain of the fractal optimal controller.

4.3 Complexity of the proposed fractal optimal control and prediction algorithms

The optimal controller discussed in Section 3 can be reduced for specific α values to a linear programming problem, which is shown to be solvable in polynomial time [36]. Please note that the difference between the fractal optimal control formulation and the conventional integer order optimal control problem is reflected only in the dynamical equation of the system, which has fractal spectrum coefficients for calculating fractional order derivative. Since the Holder exponent can take values from a predefined range a look up table can be used to compute the coefficients corresponding to the fractional order derivates. Of note estimation of the fractal spectrum can be done by using either fractal detrended fluctuation analysis [42] or large deviation method [41]; on-line fractal learning methods exists.

The prediction algorithm is also using only three parameters (Holder exponent and two parameters for relating the current derivative of the time series to the next value; these parameters can be estimated via wavelets and linear regression methods) to predict the future values. The linear regression model used for estimating these parameters can also be implemented as a linear program. In fact, the learning phase contructs the fractional order operators as a vector in which the elements are coefficients in equation (25). After the learning phase, the prediction is using only a constant amount of operations to predict the future values on the basis of the previous values.

4.4 Hardware Implementation for Fractal Prediction Algorithm and Fractal Optimal Controller

Hardware architecture for fractal prediction algorithm On the basis of proposed fractal model, the predicted value for coming request is obtained from previous values by the following equation:

$$\hat{X(t)} = a \sum_{j=1}^{N} \binom{\alpha}{j} \cdot X(t-j) + b \tag{11}$$

This equation can is implemented in hardware in two steps: first the $\binom{\alpha}{j}$ coefficients are computed by the hardware architecture shown in Figure 8 and then the summation is performed as a simple finite impulse response filter. The hardware architecture in Figure 8 consists of only multiplier and register and it is based on the recursive formula for $\binom{\alpha}{j}$:

$$\binom{\alpha}{j} = \frac{\alpha - j + 1}{j} \cdot \binom{\alpha}{j-1} \tag{12}$$

Hardware architecture for Optimal Fractal Controller; Because of the convex nature of our fractal controller, we can

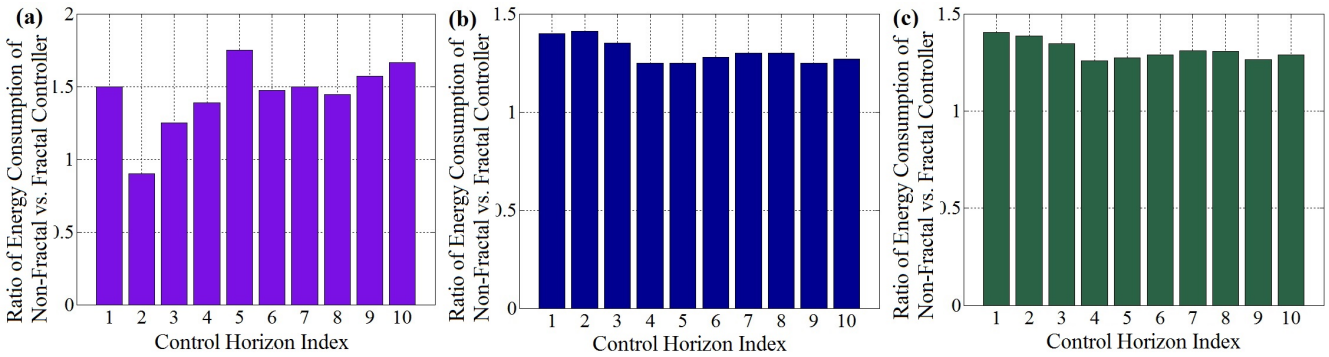

Figure 7: Energy savings due to Fractal approach in Google Cloud Computing Workload; (a). Savings due to application of fractal controller (b). Savings due to application of fractal prediction algorithm and (c). Overall savings obtained due to application of fractal prediction and control algorithm

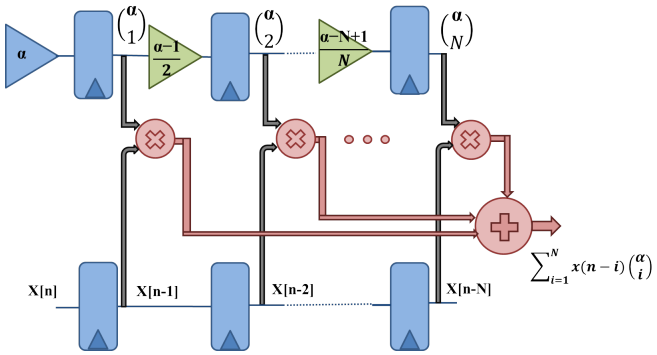

Figure 8: Hardware architecture for fractional prediction algorithm

use an Interior Point method to solve the optimization algorithm explained in Section 3. which is reduced to Linear Programming problem. We can use any state of the art linear solvers for solving e.g Simplex [37] method for solving this optimization problem. To sum up, by using all these techniques, fractal optimal controller can be synthesized efficiently in hardware and is best suited for using in real time applications.

5. CONCLUSIONS

In this paper, we introduced a mathematical model that captures the characteristics (e.g., fractal and bursty behavior) of workloads in cloud computing infrastructures. Starting from our mathematical analysis of cloud computing workload (e.g. number of requests, the remaining requests in the system) which shows a pronounced fractal behavior, we developed a wavelet based estimation methods for identifying the model parameters. We verified the accuracy of our model compared to other conventional non-fractal models on real cloud traces extracted from Google dataset [15]. Building on our novel fractal modeling of cloud workloads, we develop both a prediction algorithm and an optimal control formulation for supervision of cloud computing resources. We have shown how the fractal model is superior to conventional non-fractal model in predicting the future values with fewer parameters. We have also shown that by capturing accurately the cloud workloads dynamics makes the optimal controller engine capable of achieving significant energy savings when compared to the non-fractal model controller. We have also tested the feasibility of implementing our strategy in hardware.

6. REFERENCES

[1] B. Hayes. "Cloud Computing," *Communications of the ACM*, 2008.

[2] M. Pedram, "Energy-Efficient Datacenters," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 2012.

[3] M. Armbrust, et al. "A view of cloud computing," *Communications of the ACM*, 2010.

[4] R. H. Katz, "Tech titans building boom," *IEEE Spectrum*, 2009.

[5] J. Koomey, "Growth in data center electricity use 2005 to 2010," *Oakland, CA: Analytics Press*, August 1, 2010.

[6] H. Al-Daoud, et al. "Power-Aware Linear Programming based Scheduling for Heterogeneous Computer Clusters," *Future Generation Computer Systems*, 2012.

[7] C. K. Peng, et al. "Quantification of scaling exponents and crossover phenomena in nonstationary heartbeat time series," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 1995.

[8] J. Bi, et al. "Dynamic Provisioning Modeling for Virtualized Multi-tier Applications in Cloud Data Center," *Intl. Conf. on Cloud Computing*, 2010.

[9] M. Bjorkqvist, et al. "Opportunistic Service Provisioning in the Cloud," *Intl. Conf. on Cloud Computing*, 2012.

[10] R.Buyya, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation computer systems*, 2009.

[11] U.V. Ćataly§rek, et al. "Integrated Data Placement and Task Assignment for Scientific Workflows in Clouds," *Data-Intensive Distributed Computing*, 2011.

[12] L.Y. Chen, N. Gautam, "Server Frequency Control Using Markov Decision Processes," *INFOCOM*, 2009.

[13] Y. Chen, et al., "A First Look at Inter-Data Center Traffic Characteristics via Yahoo! Datasets,"

INFOCOM, 2011.

[14] A. Gandhi, et al. "Minimizing data center sla violations and power consumption via hybrid resource provisioning," Intl. Green Computing Conf., 2011.

[15] Google Cluster Data at: http://code.google.com/p/googleclusterdata//

[16] R. Jansen, P.R. Brenner, "Energy Efficient Virtual Machine Allocation in the Cloud," *Intl. Green Computing Conference*, 2011.

[17] M. Hadji, D. Zeghlache, "Minimum Cost Maximum Flow Algorithm for Dynamic Resource Allocation in Clouds," *Intl. Conf. on Cloud Computing*, 2012.

[18] B. Hayes, "Computing Science: Life Cycles," *American scientist*, 2005.

[19] I. Hwang et al. "Portfolio Theory-Based Resource Assignment in a Cloud Computing System," *Intl. Conf. on Cloud Computing*, 2012.

[20] Z. Liu et al. "On maximizing service-level-agreement profits," *ACM conference on Electronic Commerce*, 2001.

[21] Z. Liu et al. "Greening geographical load balancing," *SIGMETRICS*, pp. 233-244, 2011.

[22] S.T. Maguluri et al. "Stochastic Models of Load Balancing and Scheduling in Cloud Computing Clusters," *INFOCOM*, 2012.

[23] X. Meng et al. "Improving the Scalability of Data Center Networks with Traffic-aware Virtual Machine Placement," *INFOCOM*, 2010.

[24] L. Rao et al. "Minimizing electricity cost: optimization of distributed internet data centers in a multi-electricity-market environment," *INFOCOM*, 2010.

[25] M. Stillwell et al. "Resource Allocation Using Virtual Clusters," *Intl. Symp. on Cluster Computing and the Grid*, 2009.

[26] W. Shi et al. "Resource Allocation with a Budget Constraint for Computing Independent Tasks in the Cloud," *Intl Conf. on Cloud Computing Technology and Science* (CloudCom), 2010.

[27] R. Stanojevic, R. Shorten, "Distributed Dynamic Speed Scaling," *INFOCOM*, 2010.

[28] H.N. Van, et al. "Performance and Power Management for Cloud Infrastructures," *IEEE Intl. Conf. on Cloud Computing*, 2010.

[29] Y. Wang, et al. "Power Optimization with Performance Assurance for Multi-tier Applications in Virtualized Data Centers," *Intl. Conf. on Parallel Processing Workshops*, 2010.

[30] B. Whitcher et al. "Wavelet estimation of a local long memory parameter," *In Exploration Geophysics*, 2000.

[31] D. Wilcox, et al. "Probabilistic Virtual Machine Assignment," *Intl. Conf. on Cloud Computing, GRIDs, and Virtualization*, 2010.

[32] H. Xu and B. Li, "A General and Practical Datacenter Selection Framework for Cloud Services," *Intl. Conf. on Cloud Computing*, 2012.

[33] J. Yao, et al. "Dynamic Control of Electricity Cost with Power Demand Smoothing and Peak Shaving for Distributed Internet Data Centers," *Intl. Conf. on Distributed Computing Systems*,2012.

[34] L.A. Barroso and U. Holzle, "The Case for Energy-Proportional Computing," *Computer* vol. 40, pp. 33-37, 2007.

[35] X. Fan, W-D Weber and L.A. Barroso and U. Holzle, "Power Provisioning for a Warehouse-sized Computer," *Proceedings of ISCA* 2007.

[36] N. Karmarkar, "A new polynomial-time algorithm for linear programming," *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, 1984.

[37] S. Bayliss et al. "An FPGA implementation of the simplex algorithm." *Proceedings of IEEE International Conference on Field Programmable Technology*, 2006.

[38] C. Reiss, A. Tumanov, G. R. Ganger, R. H. Katz, and M.A. Kozuch, "Heterogeneity and dynamicity of clouds at scale: Google trace analysis," *Proceedings of the Third ACM Symposium on Cloud Computing (SoCC '12)* 2012.

[39] S. Di, D. Kondo, and W. Cirne, "Characterization and Comparison of Cloud versus Grid Workloads." *Proceedings of IEEE International Conference on Cluster Computing (CLUSTER)*, pp.230 - 238, 2012.

[40] Z. Liu and S. Cho, "Characterizing Machines and Workloads on a Google Cluster." *Proceedings of International Conference on Parallel Processing Workshops (ICPPW)*, pp.397 - 403, 2012.

[41] J.L. Vehel and M. Rams, "Large Deviation Multifractal Analysis of a Class of Additive Processes With Correlated Nonstationary Increments," *IEEE/ACM Transactions on Networking*, vol.21, no.4, pp.1309-1321, Aug. 2013

[42] J.W. Kantelhardt et al., "Multifractal detrended fluctuation analysis of nonstationary time series." *Physica A: Statistical Mechanics and its Applications*, pp. 87-114, 2002.

[43] M. Ghorbani and P. Bogdan, "A cyber-physical system approach to artificial pancreas design.," *Proceedings of the International Conference on Hardware/Software Codesign and System Synthesis*, 2013.

[44] P. Bogdan, R. Marculescu and S. Jain. "Dynamic power management for multidomain system-on-chip platforms: an optimal control approach." *ACM Transactions on Design Automation of Electronic Systems (TODAES)* 2013.

[45] D. Wang, C. Ren, S. Govindan and A. Sivasubramaniam, "ACE: Abstracting, Characterizing and Exploiting Datacenter Power Demands," *Proceedings of the IEEE International Symposium on Workload Characterization*, 2013.

[46] M. Wang, T. Madhyastha, N. H. Chan, S. Papadimitrious and C. Faloutsos, "Data mining meets performance evaluation: Fast algorithms for modeling bursty traffic," *Proceedings of the IEEE International Symposium on Data Engineerings*, 2002.

[47] Z. Yunyue and D. Shasha, "Efficient elastic burst detection in data streams" *Proceedings of the IEEE International conference on Knowledge discovery and data mining*, 2003.

[48] Q. Zhang, M. F. Zhani, S. Zhang, Q. Zhu, R. Boutaba and J.L. Hellerstein, "Dynamic energy-aware capacity provisioning for cloud computing environments," *Proceedings of the 9th international conference on Autonomic computing*, 2012.