Trace-Based Analysis and Prediction of Cloud Computing User Behavior Using the Fractal Modeling Technique

Shuang Chen, Mahboobeh Ghorbani, Yanzhi Wang, Paul Bogdan, Massoud Pedram
Department of Electrical Engineering
University of Southern California
Los Angeles, the United States
{shuangc, mahboobg, yanzhiwa, pbogdan, pedram}@usc.edu

Abstract—**The problem of big data analytics is gaining increasing research interest because of the rapid growth in the volume of data to be analyzed in various areas of science and technology. In this paper, we investigate the characteristics of the cloud computing requests received by the cloud infrastructure operators. The cluster usage dataset released by Google is thoroughly studied. To address the self-similarity and non-stationarity characteristics of the workload profile in a cloud computing system, fractal modeling techniques similar to some cyber-physical system (CPS) applications are exploited. A trace-based prediction of the job inter-arrival time and aggregated resource request sent to server cluster in the near future is effectively performed by solving fractional-order differential equations. The distributions of important parameters including job/task duration time and resource request per task in terms of CPU, memory, and storage are extracted from the cluster dataset are fitted using the alpha-stable distribution.**

Keywords- cloud computing; alpha-stable distribution; fractional order calculus; Google cluster dataset.

I. INTRODUCTION

In recent years, a variety of areas have seen rapid growth in the amount of data to be recorded, analyzed, and processed [1][2]. According to [3], the United States needs 140,000 to 190,000 more workers with "deep analytical" expertise and 1.5 million more data-literate managers to deal with the data flood. One of these areas with data explosion is the Internet. The Internet has become a more and more complex system in terms of both the ever increasing user population and a number of emerging applications and services. Inevitably, a huge amount of communication data is generated in the process of interaction between different network nodes.

Cloud computing, a popular and well-developed paradigm which is usually implemented through the Internet, is our major interest in this paper (with the system framework shown in Fig. 1). Different from the prior work which focuses on exploiting the cloud infrastructure to tackle the problem of big data analytics "on" the cloud [4], we will look into the problem of data analytics "for" the cloud, i.e. the characteristics of the communications and computations within a cloud computing system, especially the requests sent from cloud users to and being processed in the cluster. Such characteristics, if studied carefully, can be used to model the users' behavior, which provides useful information for a cloud infrastructure operator to optimize the operational cost and improve the quality of service (QoS). The cloud infrastructure operators are concerned about a number of aspects of cloud users' behaviors including the task incoming rate, the amount of requested CPU/memory/storage resources, the duration of the tasks, etc. Based on these analysis results, efficient management techniques, such as server consolidation and load balancing, can be applied to achieve a desirable tradeoff between the power consumption and the processing latency, which are the two major performance metrics, and maximize the overall profit. In order to find the pattern in the aforementioned aspects, data sampled in real world rather than simulated user behavior is preferred because of the highly diverse workload profile in the real-world cloud computing system, ranging from scientific computing to software development and testing. Fortunately, Google, as a leading cloud infrastructure operator, has released a substantial cluster usage dataset [5], which can be used for a comprehensive understanding of the details of the cloud computing workload.

The work presented in this paper has two major components. First, in order to capture the dynamics of the series of incoming task, an effective prediction method is proposed to estimate the workload profile (i.e., the inter-arrival time between jobs) and the resource request in terms of CPU and memory in the cluster in the near future based on the history information. Second, the distributions of several workload-related parameters extracted from the cluster dataset (e.g. job duration, CPU resource request, etc.) are analyzed and a statistical fitting is derived for these distributions. It is worth noting that neither of the two problems is trivial. Because a large number of users share a common computing infrastructure, there is a complex mixture of different types of workload, thereby making the aggregated workload pattern difficult to predict. Also, as is pointed out in [6], the job durations and resource request per

This work is supported in part by the Software and Hardware Foundations program of the NSFs Directorate for Computer & Information Science & Engineering.

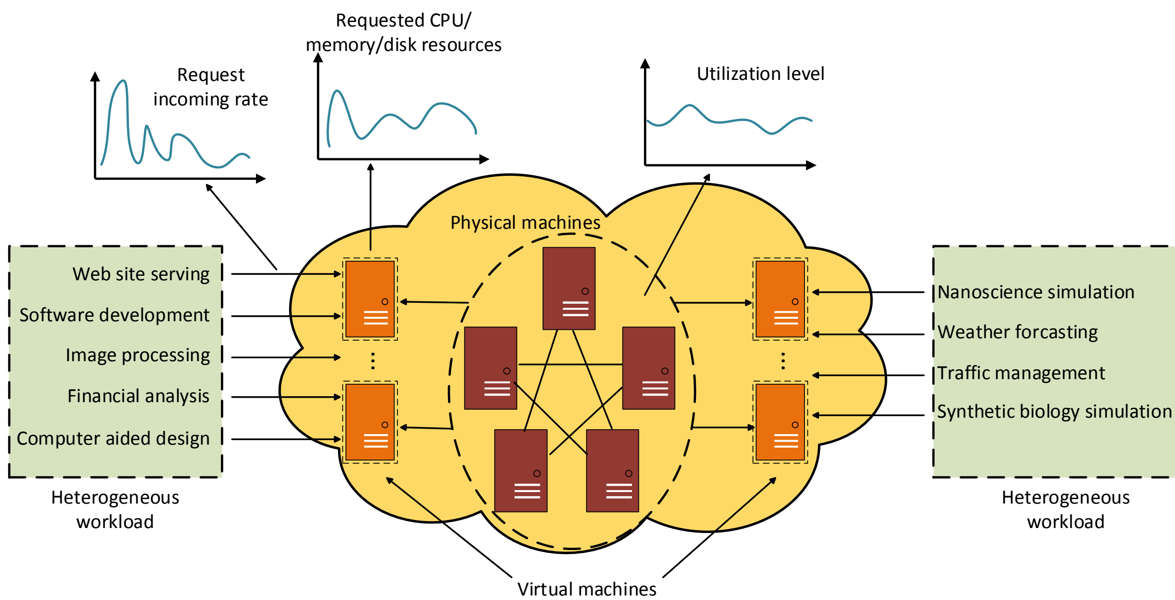

Figure 1. System framework of a cloud computing system

job form heavy-tailed distributions and cannot be accurately fitted by common statistical distributions like lognormal, Weibull, or power law distributions. However, we make the observation that the workload profile in a cloud computing system exhibits the characteristics that resemble some other physical processes, such as heart beat of a human being [7] or the cumulative concentration of cloud condensation nuclei (CCN) collected via a CCN spectrometer [8], which are also known as the workload of a cyber-physical system (CPS) [9]. Two examples of such characteristics are *self-similarity*, *fractality*, and *non-stationarity*. Self-similarity is the property that a series looks the same under the magnification operation at different scales, fractality is the property that a structure/process possesses non-integer fractal dimensions, and non-stationarity is the property that the distribution and statistical moments of a random variable do not remain the same over time. In light of these properties, we use the *fractal modeling* method which is effective in CPS applications. The distributions of extracted parameters are fitted using the alpha-stable distribution [10], and the prediction is performed by solving fractional-order differential equations.

The rest of this paper is organized as follows: Section II presents a review of the related work; Section III shows an overview of the Google cluster dataset used in this paper; the prediction and distribution fitting results are elaborated Section IV and Section V, respectively; and the last section is the conclusion.

II. RELATED WORK

Since the concept of "big data" has been brought up, general discussion regarding the benefits, challenges, and drawbacks are made in a series of research articles [11][12]. Also, a number of models and techniques have been proposed to address the issue of big data storage and processing. For instance, a general model for big data computing and communication called DOT is proposed in [13], and an optimized MapReduce framework for a specific processor is presented in [14].

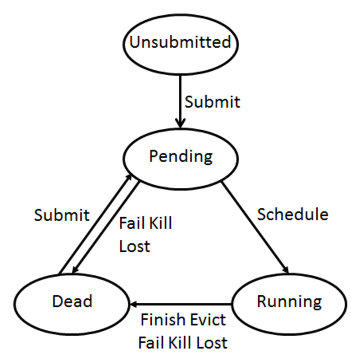

Figure 2. State transition diagram of a job/task in the cluster

There are also researches based on the Google cluster dataset. Reiss et al. [6] discusses the heterogeneity and the dynamicity of the workload on the cloud and denies the usage of some popular simplified assumptions including Poisson arrival rate and Gaussian distribution for the task duration. Di et al. [15] 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. Liu et al. [16] focuses on the frequency and pattern of machine maintenance events, job and task level workload behavior, and how the overall resource on the cluster is used. Finally, Zhang et al. [17] addresses the dynamic capacity provisioning problem that minimizes the total energy cost subject to a specific delay constraint.

The study of self-similarity property (first proposed in [18]) in the context of computer networks, which is also observed from the workload characteristics from the Google cluster dataset as will be discussed in this paper, can be traced back to [19][20], which presents the Ethernet traffic is self-similar. The study was extended by the authors of [21] to show that the traffic in the World Wide Web (WWW) is also self-similar. Moreover, the self-similarity in the topology of the Internet is analyzed in [22].

In contrast to the prior work that focus on the Google cluster dataset, we consider a comprehensive statistical analysis of the characteristics of the workload in a cloud computing system and demonstrate that alpha-stable distribution provides a good statistical model for the distribution of the job/task duration and the resource request per task. In addition, we exploit the compact representation of the workload in a cloud computing system provided by our fractal analysis to the dynamics of parameters such as the aggregated CPU/memory requests per time slot, which can further enable more efficient optimization strategies.

III. An Overview of Google Cluster Dataset

The Google cluster dataset [5] released in 2011 is measured on a heterogeneous 7000-machine server cluster on a 29-day period involving 672,075 jobs and more than 48 million tasks. The whole dataset is partitioned into six families, namely, machine events, machine attributes, job events, task events, task usage, and task constraints, which covers a wide range of information regarding the server cluster the incoming job/task sequence. The machine events show the addition, change, and removal of machines in the cluster, as well as the platform and available CPU/memory resources of each machine. The machine attributes includes other attributes that can be considered as task constraints. The job events and task events dataset record the state transition of each job/task (the state specification is discussed later). And the task usage dataset contains the mean and maximum usages of resource, i.e., CPU, memory, disk, and I/O, of every task measured in each five-minute time interval. Our focus in this paper will be on job-related and task-related information.

According to the cluster trace, a job, which contains one or more tasks, is the minimum unit of any user request received by the cloud. Once the job is submitted (i.e., received by the server cluster), however, different tasks within it can be scheduled and executed separately among different servers. A job finishes execution only after all its tasks have finished executions. Jobs and tasks share the same state transition diagram, which is shown in Fig. 2. As can be seen from the diagram, once a job/task is submitted, it will wait in the pending state to be scheduled to a server machine for execution by the cluster scheduler, after which it will enter the running state. After finishing execution, the job/task will be put into the dead state. It will also enter the dead state in the following cases: the job/task (i) is evicted to release resource for other tasks with higher priority, (ii) has a failure

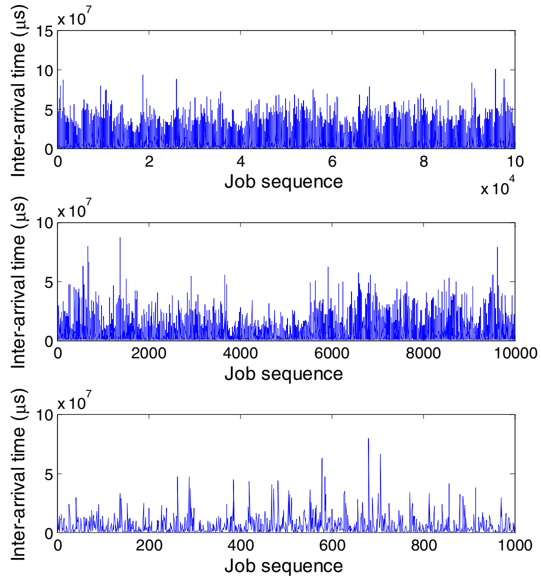

Figure 3. Job inter-arrival time series plotted under different scales

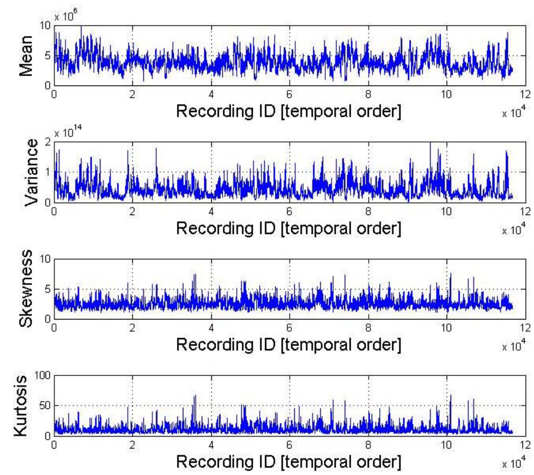

Figure 4. Statistical moments to the fourth order of the job inter-arrival time series

at execution time (or in rare cases, while pending), (iii) is canceled by the user, or (iv) is terminated abnormally for other reasons. A job/task in the dead state can be submitted again if it does not finish execution earlier. Please note that in this paper, the duration of a job/task is calculated as the difference of the time when it enters the “submitted” state and the time when it enters the “dead” state. Although the jobs/tasks that are put back into the scheduling queue retain their identifiers, we treat the resubmitted job/task the same as a new coming one, since this kind of jobs/tasks no longer require any attention or resource allocation from the cluster before the resubmission happens.

IV. Dynamic Prediction Method and Results

To investigate the dynamics of the jobs and resource requests arriving at the cluster, which is important for workload profiling in a cloud computing system, and further

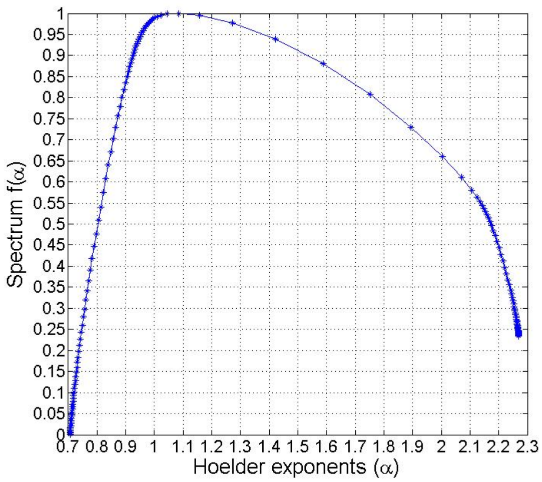

Figure 5. Multi-fractal spectrum of the job inter-arrival time series

develop an effective prediction method, we extract the job inter-arrival time and the aggregated resource request from the Google cluster trace. The job inter-arrival time sequence is generated by processing the records in the dataset of job events in temporal order and calculating the time difference between one job submission and the previous one. The aggregated CPU and memory requests of the cluster are calculated for each five-minute time slot.

A. *Complex nature of underlying processes in a cloud computing system*

As is stated earlier, the dynamics of the job arrival or the resource requests sent to the cluster are complex and challenging to model. For instance, the job arrival process cannot be simply modeled as a Poisson process or other simple stochastic processes [6]. Besides, the process exhibits similar properties to some CPS applications in terms of *self-similarity* and *non-stationarity* [9]. Fig. 3 shows the plot of job inter-arrival time series under different scales (with one hundred thousand, ten thousand, and one thousand sample points, respectively). One can observe that the job inter-arrival time series not only exhibits rich variability, but also has some degree of self-similar behavior. The plots of mean value, variance, skewness (defined as the third standardized moment), and kurtosis (defined as the fourth standardized moment) of different segments of the job inter-arrival time series are shown in Fig. 4. As can be seen from the figures, these statistical moments do not remain stable even for a short segment in the series, which indicates that the process cannot be characterized as stationary or quasi-stationary. The complex behavior of the job arrival process implicates that the process is not Gaussian and the theory of linear time-invariant (LTI) system is not applicable to model this process. The observed variability of higher order moments and the self-similarity suggest that the workload in a cloud computing system may possess a multi-fractal signature. Consequently, we estimate the multi-fractal spectrum[23][24] as shown in Fig. 5. One can see from the figure a wide distribution of fractal dimensions ranging from 0.7 to 2.3 centering at around 1.05. The complex multi-fractal behavior of the job inter-arrival time vector implies the existence of long range dependency. In other words, the job inter-arrival times after two different jobs have some degree of dependence even if they have a large number of other jobs in between.

B. *Fractal modeling based prediction method*

For the server cluster to know when to turn off some servers for power saving or reserve some resource for the imminent job incoming burst, an accurate estimation of the number of incoming tasks and the amount of available resources in the near future is crucial. To address the long range memory property and the time dependent nature of the job arrival process and the resource request, we propose a fractional order differential equation prediction model with time dependent parameters.

The α-th order derivative of a function $f(x)$ with any real value α, denoted by $D^{\alpha}f(x)$ is defined as follows:

$$D^{\alpha}f(x) = \frac{1}{\Gamma(m-\alpha)} \cdot \frac{d^m}{dx^m}\int_0^x (x-y)^{m-\alpha-1}f(y)dy \qquad (1)$$

where $m = \lceil \alpha \rceil$ and $\Gamma(\cdot)$ is the gamma function. When α is equal to a positive integer, Eqn. (1) will reduce to the conventional definition and derivatives. Since we are interested in modeling discrete sequences rather than continuous functions, we use a binomial approximation of Eqn. (1), which is shown as follows [25]

$$D^{\alpha}N(k) \cong \sum_{j=0}^{k}(-1)^j \cdot \binom{\alpha}{j} \cdot N(k-j) \qquad (2)$$

where $N(k)$ is the sequence we are interested in (aggregated resource request sent to the cluster in a time slot or the time period between two consecutive incoming jobs), and $\binom{\alpha}{j}$'s are binomial coefficients.

Using the fractional order derivatives as defined in Eqn. (2), we propose the following model

$$D^{\alpha(k)}N(k) = a(k)N(k) + b(k) \qquad (3)$$

where $\alpha(k)$, $a(k)$, and $b(k)$ are all time-dependent parameters. For each k, the value of $N(k)$ can be predicted using the values of $N(0)$ up to $N(k-1)$ by applying the following steps: (i) we estimate the order of fractional derivative, $\alpha(k)$, through a wavelet scaling analysis approach inspired from [26]; (ii) a low dimension linear regression problem is solved to find the value for parameters $a(k)$ and $b(k)$; (iii) By combining Eqn. (2) and (3) and plugging in all the known values, the value of $N(k)$ is obtained. For the purpose of reducing computation complexity, the first few binomial term rather than the full binomial expansion can be used to calculate the value of $D^{\alpha}N(k)$ as in Eqn. (2). By following this approach, we construct a compact model for the workload in a cloud computing system with few parameters.

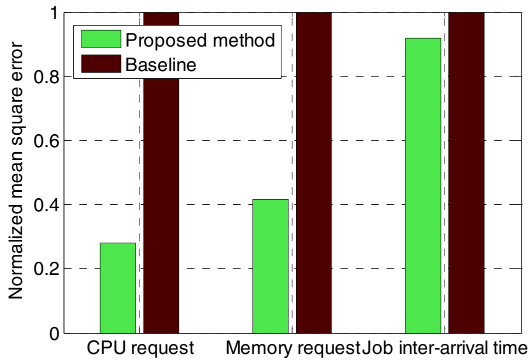

Figure 6. Comparison of prediction accuracy

C. *Prediction results and discussion*

The proposed fractional differential equation method is compared with an auto-regressive (AR) predictor [27] up to the order of 16 as the baseline. The comparison of prediction accuracy in terms of mean square error (MSE) is shown in Fig. 6. The errors of the baseline predictor are normalized to 1.

For aggregated requests of CPU and memory resource, significant improvement is achieved as can be seen from the figure. The MSE reduction is 72% and 59% in the case of CPU request and memory request, respectively. For the prediction of job inter-arrival time, the proposed method achieved 8% reduction of mean square error compared to the AR predictor. But at some sampling points where the job inter-arrival time is much larger (by several orders of magnitude) than the neighboring samples, the prediction accuracy can have significant degradation.

V. Statistical Distribution Fitting Method and Results

Apart from the job inter-arrival time and aggregated resource request of the cluster, we also extract the vector of job duration, task duration, and resource request per task in terms of CPU, memory, and disk storage. However, when applying the same prediction approach proposed in Section IV to these vectors, we cannot get accurate prediction results. The main reason is that the dependence between different elements in these vectors is relatively weak or too complex to be captured by a simple fractal model, and thus the vectors behave similarly to a series of i.i.d. random variables that are drawn from a certain statistical distribution. Therefore, instead of trying to design an accurate prediction method, we turn to finding an appropriate statistical distribution that is the most effective to fit the data. Trace extraction results show that the job duration, the task duration, and the resource request per task form a heavy-tailed distribution, which means that there is no satisfactory Gaussian fitting for them. In this paper, a general class of distributions, the alpha-stable distribution, is adopted, which captures the heavy-tailed nature of the empirical distribution and provides high flexibility for statistical analysis.

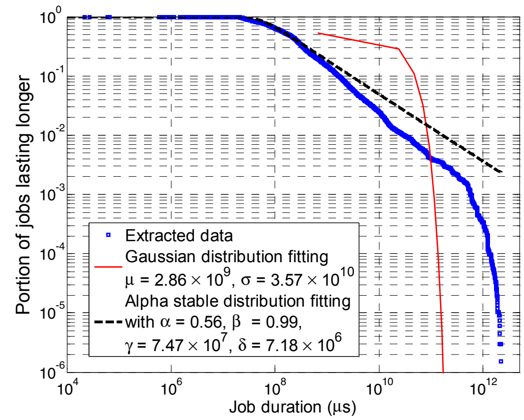

Figure 7. Statistical distribution fitting for the job duration

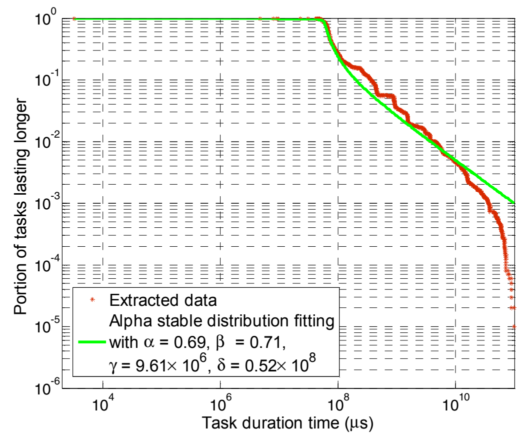

Figure 8. Statistical distribution fitting for the task duration

A. *Alpha-stable distribution*

Alpha-stable distributions do not have analytically expressible probability density functions (pdf) or cumulative distribution functions (cdf) in the general case, but can be characterized by the *characteristic functions*, $\varphi(t)$, with the form as follows

$$\varphi(t;\alpha,\beta,\gamma,\delta) = \exp[it\delta - |\gamma t|^{\alpha}(1 - i\beta \operatorname{sgn}(t)\Phi)] \tag{4}$$

where

$$\Phi = \begin{cases} \tan\left(\frac{\pi\alpha}{2}\right), & \alpha \neq 1 \\ -\frac{2}{\pi}\log|t|, & \alpha = 1 \end{cases} \tag{5}$$

$$\operatorname{sgn}(t) = \begin{cases} 1, & t > 0 \\ 0, & t = 0 \\ -1, & t < 0 \end{cases} \tag{6}$$

And the relationship between the pdf of a distribution, $f(x)$, and its characteristic function can be expressed as follows

$$f(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \varphi(t) e^{-ixt} dt \tag{7}$$

Alpha-stable distribution represents a flexible set of probability distributions. By properly setting the values of the parameters α, β, γ, and δ, one can change the shape of the pdf and cdf of an alpha-stable distribution. One can observe from the characteristic function of an alpha-stable distribution that it will reduce to some simple distributions with some specific settings of these parameters. For instance, when $\alpha = 2$, parameter β has no effect in Eqn. (4), and the distribution reduces to a Gaussian distribution with the mean of δ and the variance of $2\gamma^2$. The alpha-state distribution can be reduced to long-tail distributions also. For example, in the cases that $\alpha = 1, \beta = 0$, and $\alpha = 1/2, \beta = 1$, the alpha-stable distribution will reduce to the Cauchy distribution and the Levy distribution, respectively.

B. *Statistical fitting results and discussion*

Please note that both the Gaussian fitting and the alpha-stable distribution fitting are obtained through maximum likelihood estimation (MLE).

Fig. 7 shows the distribution fitting for the job duration, and Fig. 8 shows the distribution fitting for the task duration performed on a segment of the whole trace containing approximately 1.5 million tasks. The parameters of the alpha-stable distribution are shown in the figures. As can be seen from Fig. 7 and Fig. 8, the distributions are good fit to the corresponding data extracted from the cluster trace. In spite of the disparity that begins to appear when the job/task duration grows, we can use the fitted alpha-stable distribution to represent the real data and provide guidance for efficient resource management policies for the server cluster without any severe problem for the following reasons. First, since we only use a finite number of samples to generate the empirical survival functions, it is likely that some rare events (i.e. jobs/tasks with extremely long duration) are not captured in this process, and the proportion that long lasting jobs/tasks are actually higher than that shown in the figure, which reduces the difference between the fitted distribution and the real data. Second, if we generate random task durations according to the fitted distribution to simulate the task profile in real world, the error is within acceptable range for most of the time, since the empirical survival function only begin to decrease sharply around the first permille. And last but not the least, as opposed to the Gaussian distribution fitting results, the alpha-stable distribution fitting tends to overestimate the proportion of long-lasting tasks, which will only result in conservative resource management policies that do not cause serious problems such as violations in the service level agreement.

Fig. 9 shows the distribution fitting results for the CPU, memory, and disk storage request per task. The Google trace

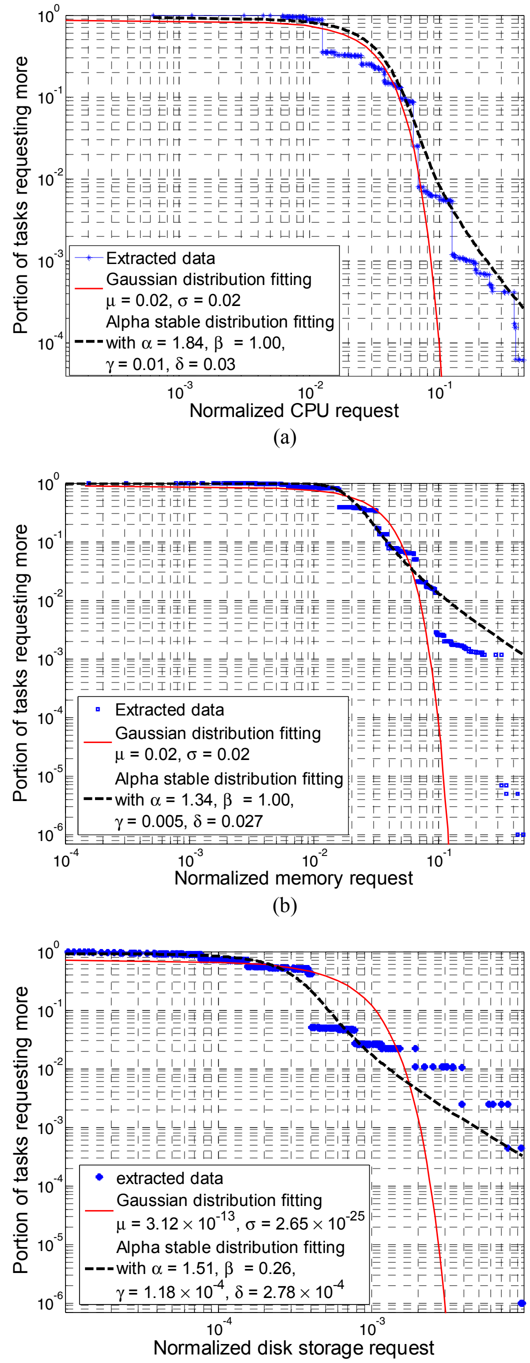

Figure 9. Statistical distribution fitting for resource request per task. (a) CPU request; (b) memory request; (c) disk storage request.

only provided the normalized values of these requests per task. The fitting is performed on a segment of the total trace which contains approximately 1.5 million tasks. As can be seen from the figure, the empirical inverted cdf curve of the resource request resource request per task is relatively unsmooth compared to that of the job duration or the task duration, which is because of the fact that the requested amount of resource is usually set manually by the users. Nevertheless, the alpha-stable distribution fits the heavy tail of the empirical survival function well. Although the Gaussian fitting for the resource requests seems acceptable for tasks requesting relatively small portion of resources (e.g. 0.05 unit of CPU or memory resource), it significantly underestimate the probability that a task request for large amount of resources, which in fact requires more attention in order to find an appropriate machine to execute it.

VI. CONCLUSION

In this paper, we investigate some important parameters extracted from the Google cluster dataset related to the workload characteristics in a cloud computing system including the job inter-arrival time, the job/task duration, the resource request per task, and the aggregated resource request sent to the cluster. Since no simple dynamic or statistical model can characterize the self-similarity and non-stationarity property that these parameters exhibit, a set of fractal modeling techniques are applied. Based on the prediction of the job inter-arrival time and the aggregated resource request sent to the cluster and the estimation of job/task duration and resource request per task in this paper, efficient resource management techniques can be further developed to benefit the cloud infrastructure operators.

REFERENCES

[1] R. E. Bryant, "Data-intensive supercomputing: The case for DISC," 2007.

[2] J. F. Gantz, "The diverse and exploding digital universe: An updated forecast of worldwide information growth through 2011," IDC, 2008.

[3] S. Lohr, "The age of big data.", *New York Times* 11, 2012.

[4] D. Agrawal, S. Das, A. E. Abbadi, "Big data and cloud computing: current state and future opportunities," In *Proceedings of the 14th International Conference on Extending Database Technology*, pp.530 - 533, 2011.

[5] " Google cluster data." [Online]. Available: https://code.google.com/p/googleclusterdata/

[6] 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.

[7] P. Ch. Ivanov, et al., "Scaling and universality in heart rate variability distributions," *Physica A: Statistical Mechanics and Its Applications*, vol. 249, no. 1, pp.587 – 593,

[8] P. Bogdan, R. Marculescu, "Workload Modeling and Related Issues for Designing Future Cyber Physical Systems," *IEEE Design and Test of Computers*, 2010.

[9] P. Bogdan, R. Marculescu, "Towards a Science of Cyber-Physical Systems Design," *Cyber-Physical Systems (ICCPS), 2011 IEEE/ACM International Conference on* , pp.99 - 108, 2011.

[10] W. Feller, An introduction to probability theory and its applications. Vol. 2. John Wiley & Sons, 2008.

[11] D. Boyd, K. Crawford, "Six Provocations for Big Data," A Decade in Internet Time: Symposium on the Dynamics of the Internet and Society, 2011.

[12] C. Lynch, "Big data: How do your data grow?" *Nature*, vol. 455, no. 7209, pp. 28-29, 2008.

[13] Y. Huai, R. Lee, S. Zhang, C. H. Xia, and X. Zhang, "DOT: a matrix model for analyzing, optimizing and deploying software for big data analytics in distributed systems," In *Proceedings of the 2nd ACM Symposium on Cloud Computing* (SOCC '11), 2011.

[14] M. Lu, L. Zhang, H. P. Huynh, Z. Ong, Y. Liang, B. He, R. S. M. Goh, R. Huynh, "Optimizing the MapReduce framework on Intel Xeon Phi coprocessor," *2013 IEEE International Conference on Big Data*, pp.125 - 130, 2013.

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

[16] Z. Liu, S. Cho, "Characterizing Machines and Workloads on a Google Cluster," *2012 41st International Conference on Parallel Processing Workshops (ICPPW)*, pp.397 - 403, 2012.

[17] Q. Zhang, M. F. Zhani, S. Zhang, Q. Zhu, R. Boutaba, and J. L. Hellerstein, "Dynamic energy-aware capacity provisioning for cloud computing environments," In *Proceedings of the 9th international conference on Autonomic computing*(ICAC '12), pp.145 - 154, 2012.

[18] B. B. Mandelbrot and J. W. Van Ness, "Fractional Brownian motions, fractional noises and applications," *SIAM Review*, vol. 10, pp. 422 - 437, 1968.

[19] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of Ethernet traffic," In *Proceedings on Communications architectures, protocols and applications* (SIGCOMM '93), pp. 183 - 193, 1993.

[20] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of Ethernet traffic (extended version)," *Networking, IEEE/ACM Transactions on* , vol.2, no.1, pp.1,15, 1994.

[21] M. E. Crovella, and A. Bestavros, "Self-similarity in World Wide Web traffic: evidence and possible causes," *Networking, IEEE/ACM Transactions on* , vol.5, no.6, pp.835 - 846, 1997.

[22] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the Internet topology," In *Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication* (SIGCOMM '99), pp. 251 - 262, 1999.

[23] B. B. Mandelbrot, A. Aharony, and J. Feder, Fractals in Physics: Essays in Honour of Benoit B. Mandelbrot : Proceedings of the International Conference Honouring Benoit B. Mandelbrot on His 65th Birthday, Elsevier, 1990.

[24] "FracLab." [Online]. Available: http://fraclab.saclay.inria.fr/

[25] I. Petráš, Fractional-Order Nonlinear Systems: Modeling, Analysis and Simulation, Springer, 2011.

[26] B. Whitcher, M.J. Jensen, "Wavelet estimation of a local long memory parameter," *Exploration Geophysics*, pp.94 - 103, 2000.

[27] E. J. Hannan, Multiple time series, Wiley, 1970