Reliable modeling of CPU usage in an office worker environment

Hugo Lewi Hammer, Anis Yazidi and Kyrre Begnum

Abstract

In this paper, we present a novel and realistic model for modelling CPU usage of virtual machines on a cloud. The model is based on considering the CPU consumption of a virtual machine as a one generated from a Hidden Markov Model (HMM). The model assumes that the hidden layer (Markov chain) is inhomogeneous and depends on the time of day. In addition the model assumes that the observations follow an autoregressive process. The deviations from standard HMMs are motivated by the properties of real CPU consumption data. The HMM model replicate the properties of the real CPU consumption data in a very realistic way and outperform both AR(1) and AR(2) models in predicting time ahead CPU consumption.

1 Introduction

Use of cloud computing is increasing in a tremendous speed as the use of virtual machines have exploded these last years. More companies are centralizing their resources to data centers to get uninterrupted power, better security, expansion opportunities and availability. The number of data centers has increased with 56 percent worldwide from 2005 to 2010 [4] and they are installing more hardware to handle the rapidly increasing demand. Cloud computing is now consuming more electricity every day than India [2] and Google alone is consuming the same amount of energy as the Norwegian capital [6, 7].

Even though most cloud technologies and virtual environments support the technical building blocks of dynamic power-savings, the remaining part is that of the algorithms by which to automate the process.

In this paper we study a case where the virtual machines on a cloud are used by office workers. A typical example could be a large company that use a cloud to provide resources to the office workers of the company. In order to tackle the problem of fast dynamics as described above, we model resource consumption of CPU using a generalized version of the Hidden Markov model (HMM). This paper will show that the model accommodate the main properties of CPU usage.

2 Related work

The fact that the growth of data centers has become a considerable environmental challenge and has high operational power cost has motivated a great deal of research and development on modeling resource usage.

In [8], the authors used elements from the theory of Model Predictive Control(MPC) to find the optimal control policy for dynamic capacity provisioning. A system that controls the number of active servers in a data center for energy savings. The solution aims to find a trade-off between energy savings and capacity reconfiguration cost. The framework is an initial step towards building a full-fledged management system. RSOM [1] stands for Recurrent Self-Organizing Map and is a module for making an energy-efficient self-provisioning approach for cloud resource management. RSOM was based on an unsupervised predictor model in the form of an self-organizing map that predicted the user load after historical usage.

HMMs have been used in many fields of research like speech recognition, sachine translation, gene prediction, protein folding and environmental and climate research. There are also examples in cloud computing. Khan et al. [3] clusters VMs with correlated temporal resource usage and apply a HMM to characterize the temporal correlations in the discovered VM clusters and to predict variations of workload patterns. We have not found any research paper using an inhomogeneous HMMs to model resource usage in cloud compting as proposed in this paper.

3 Properties of CPU consumption over time

To be able to construct a good consolidation algorithm, we need an understanding of the properties of resource consumption for users. We logged the CPU consumption of a typical (hard working) office worker every fifth minute for 15 working days. The data is shown in Figure 1, where the three upper panels show

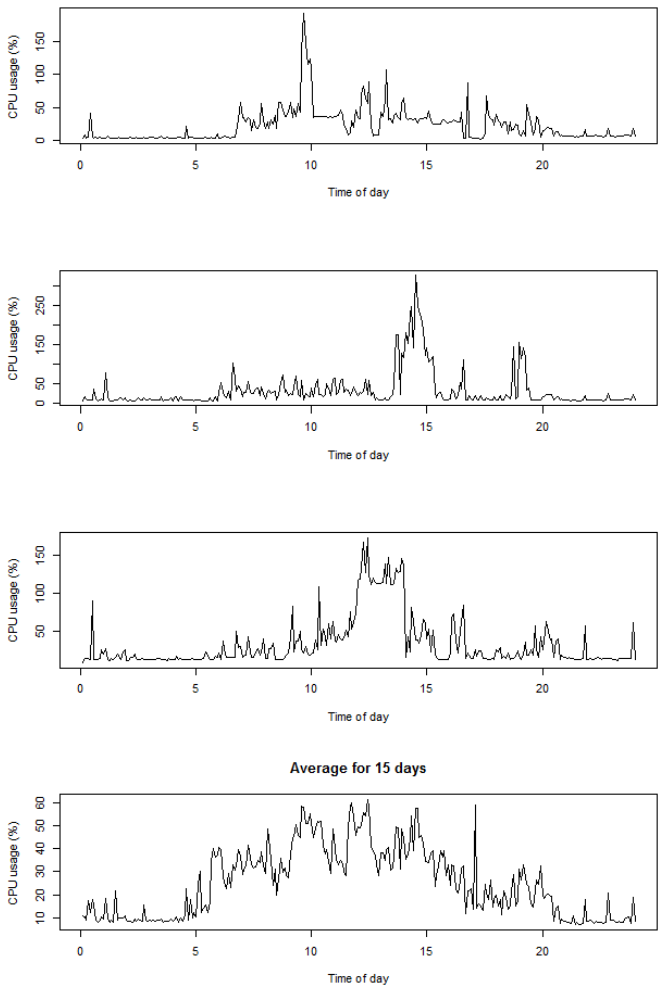

Figure 1: CPU consumption for a typical office worker. Three upper panels: CPU consumption for three arbitrary days. Bottom panel: Average CPU consumption over the 15 working days. 100% CPU consumption is equivalent to one CPU kernel working on full activity.

CPU consumption for three arbitrary days, and the bottom panel the average CPU consumption over the 15 working days we monitored the office worker. We will use the data to develop a suitable statistical model for such data. We can do the following observations from the data.

A. The office worker have high activity during the day starting from about 5:00 AM in the morning at the earliest and ending around 8:00 PM (20:00) in the evening. The highest CPU consumption on average is between 10:00 AM and 3:00 PM (15:00). There are almost no activity before 5 AM and after 8 PM.

B. It is also a clear dependence between subsequent observations (the autocorrelation is larger than zero).

4 Statistical model for CPU consumption

In this section we present a suitable statistical model for the CPU data described in Section 3.

4.1 Hidden Markov models

Let X_t be a discrete stochastic variable with possible outcomes $\{0, 1, \ldots, K\}$ representing the states of a Markov chain at time point $t \in \{1, 2 \ldots, T\}$. The distribution of an other stochastic variable Y_t

depends on the state of X_t, $Y_t \sim P(Y_t|X_t)$. Given $X_1, X_2, \ldots, X_T$, we assume that $Y_t, t = 1, 2, \ldots, T$ are independent. Overall the model can be written as

$$P(X_{1:T}, Y_{1:T}) = P(X_1)P(Y_1|X_1) \prod_{t=2}^{T} P(X_t|X_{t-1})P(Y_t|X_t) \tag{1}$$

where $X_{1:T} = X_1, X_2, \ldots, X_T$ and $Y_{1:T} = Y_1, Y_2, \ldots, Y_T$.

The conditional independence of $Y_{1:T}$ given $X_{1:T}$ in (1) is often unrealistic in many applications. One possible generalization is to let Y_t depend on the previous states, e.g. change $P(Y_t|X_t)$ with $P(Y_t|X_{t-1:t}, Y_{t-1})$ in (1). One example could be that observations are generated from an autoregressive model of lag one

$$\begin{aligned} Y_t &= m(X_t) + a(X_t)(Y_{t-1} - m(X_t)) + \epsilon_{1,t}(X_t), \\ &\qquad\qquad\qquad\qquad \text{if } X_{t-1} = X_t \\ Y_t &= m(X_t) + \epsilon_{2,t}(X_t), \quad \text{if } X_{t-1} \neq X_t \end{aligned} \tag{2}$$

where the time series depends on the state of the underlying Markov chain.

The models above can efficiently be evaluated using the Forward Backward algorithm.

4.2 Hidden Markov model for CPU consumption

We now apply a Hidden Markov model to the CPU consumption data. The CPU consumption data for office workers show as expected a periodicity of 24 hours (Figure 1). We assume that the observations for different days are independent outcomes from the same model. We assume that the hidden Markov chain have to states

- The user is using the computer at time t (active), $X_t = 1$. From Figure 1 we see that this is typically from the morning to the afternoon.
- The user is not using the computer at time t (inactive), $X_t = 0$. This is typically in the evening and the night.

For Hidden Markov models the far most common is to assume that the transition probabilities of the hidden Markov chain is constant over time. This is not a realistic assumption for this application. For example it is more likely to go from inactive to active state in the morning than in the evening or from active to inactive state in the evening than in the morning. Thus we assume a time dependent (inhomogeneous) Markov chain with transition matrix denoted as

$$\mathcal{P}_t = \begin{bmatrix} P_t^{00} & P_t^{01} \\ P_t^{10} & P_t^{11} \end{bmatrix}$$

where P_t^{ij} is the probability of going from state i to j in time step t. Since $P_t^{01} = 1 - P_t^{00}$ and $P_t^{10} = 1 - P_t^{11}$, the transition matrix consists of two free parameters in each time step. To reduce the number of unknown parameters we assume that the transition probabilities are related through some functions. The following functions turned out to perform well in our experiments

$$\begin{aligned} P_t^{00} &= \text{logit}^{-1}(\gamma_{00} + \gamma_{01}t + \gamma_{02}t^2) \\ P_t^{11} &= \text{logit}^{-1}(\gamma_{10} + \gamma_{11}t + \gamma_{12}t^2) \end{aligned} \tag{3}$$

where the γ's are parameters that will be estimated from CPU consumption data. Using a parabola within the inverse logit function, we are able to model that a user during a 24-hour cycle go from a high probability of being inactive (night) to a high probability of being active (e.g. working hours) and back again to being inactive (evening/night). Using (3), the number of unknown parameters are reduced from $2(T-1)$ to six.

Next we go to the model assumptions for the observations. Inspecting the CPU data in Figure 1 the conditional independence between subsequent observations given the hidden state does not seem realistic and thus we use the generalized version of the HMM based on autoregressive processes of lag 1, see (2).

We may not have many days with observations which results in uncertainty in the estimation of the parameters. We deal with uncertainty by casting the problem into a Bayesian framework. We use wide (non-informative) uniformly distributed prior distributions. We assume that a priori all the

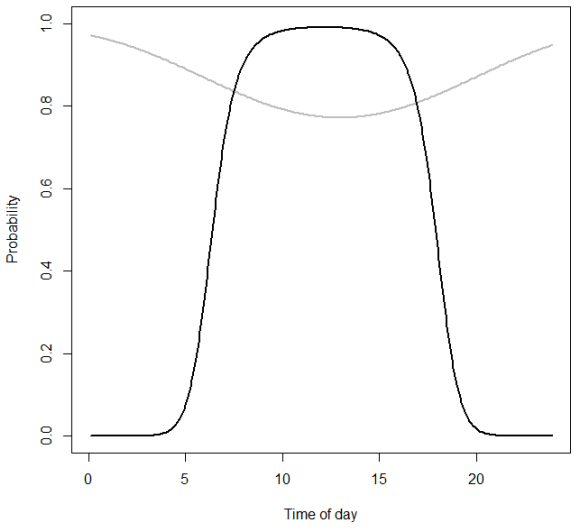

Figure 2: Estimated curves for P_t^{00} (gray curve) and P_t^{11} (black curve).

hyperparameters are independent. Further we assume that we have observations for D days (24-hour cycles) and that the observations for each day are independent given the hyperparameters. It is then straight forward to set up the posterior distribution for the unknown parameters. We evaluate the posterior distribution, i.e. estimate the unknown parameters, using Markov chain Monte Carlo (McMC) simulation [5].

5 Experiments

In the rest of the paper we denote the model presented in Section 4.2 CPU HMM. Figure 2 shows the estimated curves for P_t^{00} and P_t^{11} in equation (3) based on the samples from the MH algorithm. We see that the estimated transition matrix based on P_t^{00} and P_t^{11} is highly inhomogeneous as a function of time showing the importance of not modeling the hidden Markov chain as homogeneous. We see that in the night it is a high probability of staying in the inactive state (P_t^{00}) and a very low probability of staying in the active state (P_t^{11}). As the morning approaches P_t^{00} decreases while P_t^{11} increases rapidly which means that a transition from inactive to active state becomes very likely. In the afternoon we observe the opposite and a transition from active to inactive state becomes more and more likely. It is very nice to observe how well the given model and simulation algorithm are able to automatically separate the resource consumption in clear active and inactive states still using wide prior distributions with no information on how to separate in active and inactive state.

Figure 3 shows three independent realization from the CPU HMM model and the average of 15 realization. The black lines at the bottom of each panel show at which time intervals the underlying Markov chain were in the active state for these realizations. Because of the inhomogeneity of Markov chain we see that it is far more likely to be in active state during working hours compared to the other times of the day. Comparing Figure 3 with Figure 1, we see that the CPU HMM replicate the properties of the real CPU usage data in an impressive way.

The prediction performance of the CPU HMM was compared to the prediction performance of AR(1) and AR(2) time series models. The results show that the CPU HMM outperforms both of these commonly used models.

6 Closing remarks

In this paper we present an inhomogeneous Hidden Markov model (HMM) that replicate the properties of real CPU consumption data in a very realistic way. Our experiments show that the model has good prediction properties and out-competes both AR(1) and AR(2).

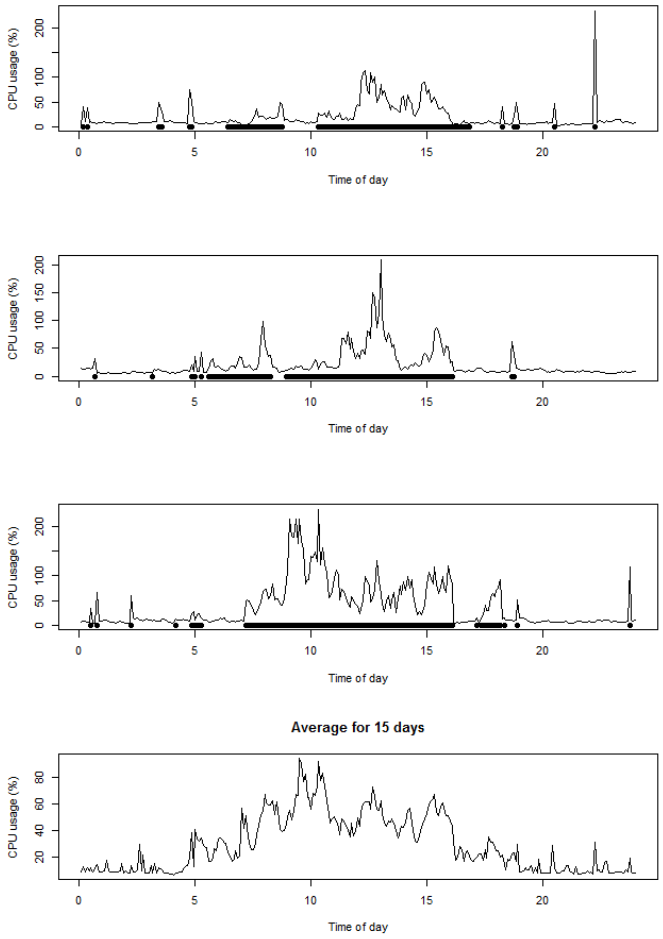

Figure 3: Three upper panels: Three arbitrary realizations from the CPU HMM. Bottom panel: Average of 15 realizations. 100% CPU usage is equivalent to one CPU kernel working on full activity. The black lines at the bottom of each panel show at which time intervals the underlying Markov chain were in the active state for these realizations.

References

[1] Hanen Chihi, Walid Chainbi, and Khaled Ghedira. An energy-efficient self-provisioning approach for cloud resources management. *ACM SIGOPS Operating Systems Review*, 47(3):2–9, 2013.

[2] Green peace. Green peace - click clean: How companies are creating the green internet. http://www.greenpeace.org/usa/Global/usa/planet3/PDFs/clickingclean.pdf, mar 2010. [Online; accessed April 2014].

[3] Arijit Khan, Xifeng Yan, Shu Tao, and Nikos Anerousis. Workload characterization and prediction in the cloud: A multiple time series approach. In *NOMS*, pages 1287–1294. IEEE, 2012.

[4] Jonathan G Koomey. Worldwide electricity used in data centers. *Environmental Research Letters*, 3(3):034008, 2008.

[5] Christian Robert and George Casella. *Monte Carlo Statistical Methods*. Springer Series in Statistics. Springer Science, New York, USA, 2004.

[6] Statistisk sentralbyrå. Statistikkbanken, antall boliger i oslo @ONLINE, January 2014.

[7] The New York Times. Google details, and defends, its use of electricity@ONLINE, September 2011.

[8] Qi Zhang, Mohamed Faten Zhani, Shuo Zhang, Quanyan Zhu, Raouf Boutaba, and Joseph L. Hellerstein. Dynamic energy-aware capacity provisioning for cloud computing environments. In Dejan S. Milojicic, Dongyan Xu, and Vanish Talwar, editors, *ICAC*, pages 145–154. ACM, 2012.