EVALUATING DEMAND PREDICTION TECHNIQUES FOR COMPUTATIONAL MARKETS

THOMAS SANDHOLM

KTH – Royal Institute of Technolog,
SE-100 44 Stockholm, Sweden,
E-mail: sandholm@pdc.kth.se

KEVIN LAI

Informations Dynamics Laboratory,
HP Labs, Palo Alto, CA 94304
E-mail: kevin.lai@hp.com

We evaluate different prediction techniques to estimate future demand of resource usage in a computational market. Usage traces from the PlanetLab network are used to compare the prediction accuracy of models based on histograms, normal distribution approximation, maximum entropy, and autoregression theory. We particularly study the ability to predict the tail of the probability distribution in order to give guarantees of upper bounds of demand. We found that the maximum entropy model was particularly well suited to predict these upper bounds.

1. Introduction

Large scale shared computational Grids allow more efficient usage of resources through statistical multiplexing. Economic allocation of resources in such systems provide a variety of benefits including allocating resources to users who benefit from them the most, encouraging organizations to share resources, and providing accountability [12,6,1,14].

One critical issue for economic allocation systems is predictability. Users require the ability to predict future prices for resources so that they can plan their budgets. Without predictability, users will either over-spend, sacrificing future performance, or over-save, sacrificing current performance. Both lead to dissatisfaction and instability. Moreover, the lack of accurate information precludes rational behavior, which would disrupt the operation of the many allocation mechanisms that depend on rational behavior.

There are three parts to predictability: the predictability provided by the allocation mechanism, the predictability of the users' behavior, and the predictability provided by statistical algorithms used to model the behavior. We examine the latter two. Consequently, these results are not dependent on a specific allocation mechanism and instead apply to many systems.

The goal of this paper is to examine the degree to which future demand can be predicted from previous demand in a shared computing platform. Ideally, we would use the pricing data from a heavily used economic grid system, but such systems have not yet been widely deployed. Instead, we examine PlanetLab[9], a widely-distributed, shared computing platform with a highly flexible and fluid allocation mechanism. The PlanetLab data set has the advantage of encompassing many users and hosts and having very little friction for allocation. However, PlanetLab does not use economic allocation, so we substitute usage as a proxy for pricing. Since many economic allocation mechanisms (e.g., Spawn[11], Popcorn[10], ICE[4], and Tycoon[7]) adjust prices in response to the demand, we believe that this approximation is appropriate.

We examine this data set using four different statistical prediction algorithms: histogram (Hist) approximation, maximum entropy (MaxEnt) density estimation, an autoregression (AR) time series model, and a normal (Norm) distribution model. We evaluated these algorithms by feeding them samples of usage data over a particular period of time and then measuring the error of the generated model. We then measured the error of using these models to predict future demand.

Our findings are as follows:

- MaxEnt and Norm were able to accurately model the data set over larger time periods. Maximum entropy estimation is approximately twice as accurate as a normal model because of its ability to capture skewness. Both methods are an order of magnitude more accurate than histogram approximation. The MaxEnt model is based on fitting integrals of the distribution function to statistical moments. This fit may not yield satisfactory approximations if the number of data samples in the time window investigated are too few, and we then fall back to the normal distribution approximation.

- All of the techniques produce inaccurate predictions, when trying to predict the cumulative distribution function for future demand. Autoregression has the additional disadvantage of requiring so much compute overhead that it was not able to complete some predictions. Furthermore, the AR model requires more history data to be maintained in order to retrain the prediction model to fit the current load.

• Despite inaccurate predictions of the full cumulative distribution function, MaxEnt and Norm were able to produce accurate *bounds* for demand. This is important because bounds are sufficient for users to budget. For example, if a user knows that the probability of hosts being less than $1 per host within the next week is 99%, and he needs 10 hosts, then he knows he should budget $10.

2. Prediction Algorithms

The goal of the prediction algorithms is to predict the demand for a resource based on historical data. In an economic system, the demand determines the price, which allows users to budget accurately. The general prediction model we use is summarized here.

$$P(Y \leq \frac{y-\mu}{\sigma}) = \Phi(\frac{y-\mu}{\sigma}) \tag{1}$$

$$y \leq \mu + \sigma\Phi^{-1}(p) \tag{2}$$

where y is the demand with mean μ and standard deviation σ, and Φ is the cumulative probability density function (CDF) of a normal distribution. Eq. 1 gives us a way to get a probability of a demand given its mean and standard deviation, and Eq. 2 allows us to find the demand corresponding to level of guarantee or probability.

In this work we want to remove the assumption of a normal distribution, and instead only assume an iid (independent identically distributed) distribution, and then compare the results to those obtained using the normal distribution assumption. More specifically, this means that we want to take the skewness of the distribution into consideration in our predictions. This extension is motivated by previous work on computational markets and usage behavior on the web [3] have shown that heavy-tailed distributions are common.

We evaluate three different approaches to tackle this generalization here, histogram (Hist) approximation, maximum entropy (MaxEnt) density estimation, and an autoregression (AR) time series model. The results are benchmarked against approximations used with the normal (Norm) distribution assumption, and compared to the real outcome.

The Hist approximation is based on placing sample data points in a fixed number of bins with predetermined data ranges. It therefore assumes some a-priori knowledge of the variance of the data. In our benchmarks we used 10 and 100 bins to approximate the distribution of values in a range of about 5000 distinct data values.

The MaxEnt model is based on the concept of choosing a distribution function which maximizes the entropy or randomness (or simply the unknown parameters) of a function given some characteristics such as statistical moments. This idea was first articulated by E.T. Jaynes in [5]. Cover and Thomas [2] then proved that all functions maximizing the entropy of a distribution are of a general form. For example, given the following constraints of the three moments about the origin μ_1, μ_2, μ_3

$$\int_{-\infty}^{\infty} f(x)dx = 1, \int_{-\infty}^{\infty} xf(x)dx = \mu_1, \int_{-\infty}^{\infty} x^2 f(x)dx = \mu_2, \int_{-\infty}^{\infty} x^3 f(x)dx = \mu_3$$

then the distribution function that maximizes the entropy has the form

$$f(x) = e^{\lambda_0 + \lambda_1 x + \lambda_2 x^2 + \lambda_3 x^3}$$

Now the problem of finding the distribution function f reduces to finding the λ parameters. Cover and Thomas suggests starting with the parameters known for a normal distribution and then "wiggle" them to find the best fit. In our implementation we performed this "wiggling" by applying the steepest descent iterative optimization algorithm described in [13]. In summary, we iteratively try to get closer to

$$\theta = \lambda_0, \lambda_1, \lambda_2, \lambda_3$$

by initializing it to the values know for a normal distribution and then assigning it subsequent values according to

$$\theta_{t+1} = \theta_t - \mathrm{H}^{-1}\mathrm{B}$$

where H is the Hessian matrix defined as

$$H_{k,j} = \int x^k x^j f(x, \theta_t)dx, 0 \leq k, j \leq 3$$

and B is the difference vector

$$B_k = \int x^k f(x, \theta_t)dx - \mu_k, 0 \leq k \leq 3$$

Note that we use the first three moments to capture the skewness of the distribution. Using more than three moments introduces irregular fluctuations which could prevent the algorithm from converging, and it also more easily runs into numerical limitations such us number overflows and round off errors.

The AR model [8] is a standard time-series model of the following form

$$X_t = \mu + \sum_{i=1}^{k} \alpha_i (X_{t-1} - \mu)$$

where μ is the measured mean in the training data, and k is the order (we used $k = 2$ in our benchmarks). The model parameters α_i are estimated by first calculating the autocorrelation vector for the training data and then solving the Yule-Walker equations. Note that the white-noise parameter has been omitted for simplicity.

Four different evaluations are performed on time series data using these techniques. First, we look at how well the summary data, such as bin density with Hist, the first three moments about the origin with MaxEnt, and the first two moments with Norm approximate the distribution described in the current period. If we have an iid distribution this should also give an indication of the possible accuracy of future predictions. Second, we look at how well predictions based on approximations of the cumulative density function in previous intervals can predict future distributions, and compare that to AR prediction results. Third, we look at how the actual distribution changes over time in the different intervals studied. Finally we look at how well the 99th percentile of the cumulative distribution function can be estimated in order to see how well guarantees can be given that the price will not exceed a certain value.

We also look at the convergence rate of the MaxEnt estimation. If it does not converge we, as previously mentioned, fall back to the Norm approach.

3. Results

We study usage time-series data, based on 5-minute snapshots of the aggregated number of PlanetLab slices allocated across the whole network. Data from two months (November-December 2005) were used. Training and future prediction horizons corresponding to predictions roughly from 2 hours to 3 days into the future were evaluated.

3.1. *Modelling*

In Figure 1 we can see that the MaxEnt approximation improves the accuracy of the CDF fit substantially compared to the normal distribution technique. SSE is the sum of the squares of the errors when plotting the

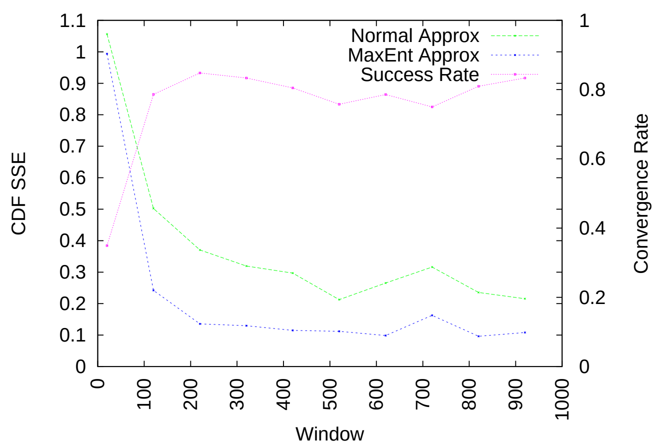

Figure 1. PlanetLab Density Approximation

CDF with a granularity of 100 data points. The windows correspond to number of 5-minute snapshots used to predict the same number of 5-minute snapshots into the future.

We can see that the MaxEnt approximation does not converge in the case of the window size 50 in more than 35% of the cases. We wanted to investigate why, and performed a correlation test on the range of the data values in the window, the standard deviation of the data, and the likelihood of convergence. We obtained correlation coefficients 0.56, and 0.55 for data range and standard deviation respectively which are significant at the 1%-level according to a t-Student test. Intuitively this may be caused by the integral calculations used in the MaxEnt fit being too short to find the underlying entropy maximizing distribution. As a clarification, convergence of the MaxEnt approximation is defined by the error when fitting to the moments expected is less then a certain value ϵ. With the PlanetLab data we saw that an ϵ of 100 worked best, but there is always a tradeoff between accuracy and convergence rate.

3.2. *Predicting the Cumumlative Distribution*

Figure 2 shows an example of an interval estimation and how the different CDF functions compare. The window size in this case was two hours. We can see that the entropy model gives a much better fit to the non-normal behavior of the curve. The histogram estimation (with 100 bins) is quite

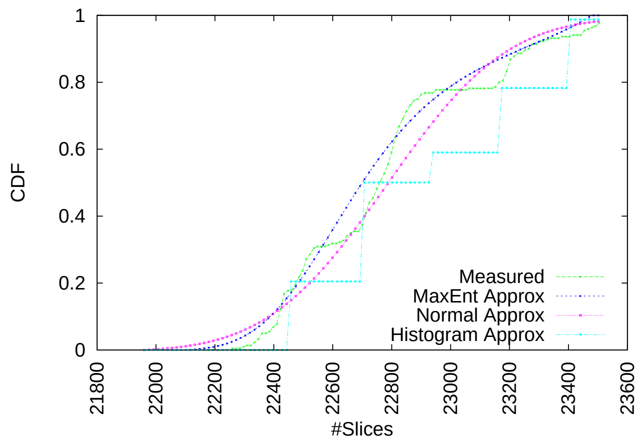

Figure 2. PlanetLab Density Approximation CDF

a coarse grained estimation, and requires more state to be maintained as opposed to just three running moments as in the entropy case.

3.3. *Predicting Bounds*

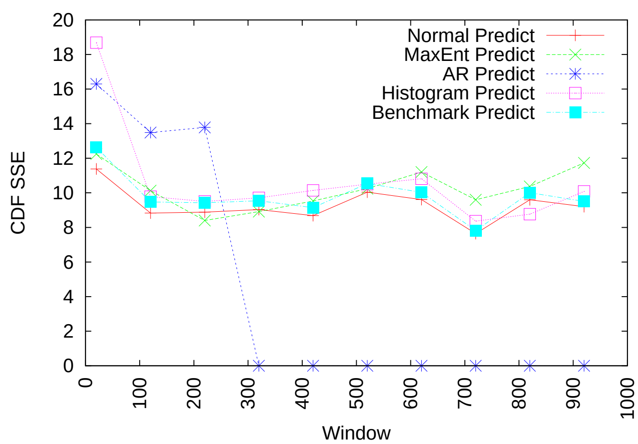

Figure 3. PlanetLab Density Prediction

A bit surprisingly we see in Figure 3 that the MaxEnt model does not produce better prediction results over time than the normal approximation. The AR curve is provided for reference. It does not make sense to use the AR model unless it predicts better than predicting the outcome of the previous period since it also requires all the data points to be kept in history. Since this is not the case for these long-interval predictions it provides no added value in this situation. Another severe limitation of AR is that it numerically due to large Matrix computations is not feasible to predict more than roughly 300 data points into the future. Note that in the graph this is shown by the AR SSE being set to 0 for window sizes greater than 300.

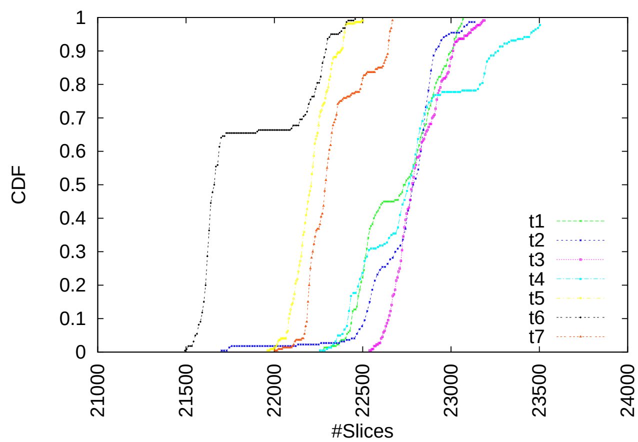

Figure 4. PlanetLab Density Variance

An explanation to why the MaxEnt model cannot benefit from its more accurate density approximations when predicting future densities can be seen in Figure 4. Each CDF in the Figure is taken in a subsequent interval so the t1 curve contains the distribution of all the data points from the start of the measurement to time t1, the t2 curve has all the data points between t1 and t2, etc. The mean point of the density moves back and forth in an unpredictable manner. Another indicator of this is the high SSE value of the benchmark prediction (predicting last periods CDF for the next) in Figure 3 (around 11) compare to the values in 1 (around 0.2).

It is then more encouraging to see that the 99th percentile MaxEnt

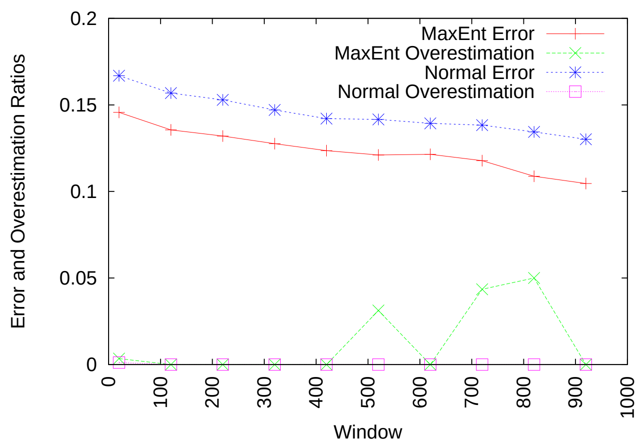

Figure 5. PlanetLab Tail Prediction

estimates in Figure 5 are more accurate than with Norm. We should also note here that the training was done on the maximum amount of history data available and not just the previous period to do more of a worst case estimation of the tail as opposed to an overall accurate one. The error presented in Figure 5 is calculated as the difference between the measured value and the approximation divided by the measured value.

4. Conclusions

Although the statistical prediction algorithms that we examine here were not able to accurately predict future demand in the PlanetLab data set, we found that the MaxEnt algorithm was able to accurately predict bounds on future demand.

Some areas for future work are to examine the performance of MaxEnt in a live system and for systems with different applications and user behaviors than PlanetLab. Ultimately we hope to examine the performance of the algorithms in a live economic Grid system.

Given the fluidity of PlanetLab usage and the lack of a pricing mechanism to moderate usage, the accuracy of the MaxEnt algorithm gives us optimism that prediction algorithms will be accurate in real economic systems. We believe that this will ultimately lead to more stable, more economically efficient systems.

References

1. R. Buyya, D. Abramson, and S. Venugopal. The Grid Economy. *Proceedings of the IEEE, Special Issue on Grid Computing*, 93(3):479–484, March 2005.
2. T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., 1991.
3. C. A. Cunha, A. Bestavros, and M. E. Crovella. Characteristics of WWW client-based traces. Technical Report TR-95-010, Boston University Department of Computer Science, Apr. 1995. Revised July 18, 1995.
4. David C. Parkes and Ruggiero Cavallo and Nick Elprin and Adam Juda and Sebastien Lahaie and Benjamin Lubin and Loizos Michael and Jeffrey Shneidman and Hassan Sultan. ICE: An Iterative Combinatorial Exchange. In *Proceedings of the ACM Conference on Electronic Commerce*, 2005.
5. E. Jaynes. Information Theory and Statistical Mechanisms. *Physics Review*, 106:620–630, 1957.
6. L. V. Kale, S. Kumar, M. Potnuru, J. DeSouza, and S. Bandhakavi. Faucets: Efficient resource allocation on the computational grid. In *ICPP '04: Proceedings of the 2004 International Conference on Parallel Processing (ICPP'04)*, pages 396–405, Washington, DC, USA, 2004. IEEE Computer Society.
7. K. Lai. Markets are Dead, Long Live Markets. *SIGecom Exchanges*, 5(4):1–10, July 2005.
8. L. Ljung. *System Identification: Theory for the User*. Prentice Hall, December 1998.
9. L. Peterson, T. Anderson, D. Culler, , and T. Roscoe. Blueprint for Introducing Disruptive Technology into the Internet. In *First Workshop on Hot Topics in Networking*, 2002.
10. O. Regev and N. Nisan. The Popcorn Market: Online Markets for Computational Resources. In *Proceedings of 1st International Conference on Information and Computation Economies*, pages 148–157, 1998.
11. C. A. Waldspurger, T. Hogg, B. A. Huberman, J. O. Kephart, and W. S. Stornetta. Spawn: A Distributed Computational Economy. *Software Engineering*, 18(2):103–117, 1992.
12. R. Wolski, J. S. Plank, T. Bryan, and J. Brevik. G-commerce: Market formulations controlling resource allocation on the computational grid. In *IPDPS '01: Proceedings of the 15th International Parallel and Distributed Processing Symposium (IPDPS'01)*, page 10046.2, Washington, DC, USA, 2001. IEEE Computer Society.
13. X. Wu and T. Stengos. Partially adaptive estimation via the maximum entropy densities. *Econometrics Journal*, 8(3):352–366, 2005.
14. L. Xiao, Y. Zhu, L. M. Ni, and Z. Xu. Gridis: An incentive-based grid scheduling. In *IPDPS '05: Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05) - Papers*, page 65.2, Washington, DC, USA, 2005. IEEE Computer Society.