Survey of Grid Resource Monitoring and Prediction Strategies*

Liang Hu [1], Xiaochun Cheng [2], Xilong Che [1]
[1] *College of Computer Science and Technology, Jilin University.*
[2] *School of Computing Science, Middlesex University.*
hul@jlu.edu.cn, x.cheng@mdx.ac.uk, chexilong@jlu.edu.cn
doi:10.4156/ijiip.vol1. issue2.8

Abstract

This literature focuses on grid resource monitoring and prediction, representative monitoring and prediction systems are analyzed and evaluated, then monitoring and prediction strategies for grid resources are summarized and discussed, recommendations are also given for building monitoring sensors and prediction models. During problem definition, one-step-ahead prediction is extended to multi-step-ahead prediction, which is then modeled with computational intelligence algorithms such as neural network and support vector regression. Numerical simulations are performed on benchmark data sets, while comparative results on accuracy and efficiency indicate that support vector regression models achieve superior performance. Our efforts can be utilized as direction for building online monitoring and prediction system for grid resources.

Keywords: *Grid Resource, Monitoring Strategy, Multi-step-ahead Prediction, Neural Network, Support Vector Regression*

1. Introduction

The allocation of the resources and the scheduling of tasks are basic problems in grid computing, and there is no doubt that resource performance is the most influencing factor within such area [1]. The performance information of grid resources are mainly achieved by means of two mechanisms: monitoring and prediction. Grid resource monitoring aims to acquire the status, distribution, load as well as the fault situation of the resources in grid environment by means of monitoring methods. While grid resource prediction aims to handle the variation principles and running traces of grid resources by means of modeling and analyzing on historical monitoring data. In a word, monitoring can provide historical information and the current information, while prediction provides future variation information. These two mechanisms are supplement to each other. Grid resource monitoring and prediction are inevitable in grid computing system. The grid needs a large amount of monitoring and prediction data:

- to carry on performance analysis, service control, bottleneck elimination and fault diagnosis;
- to provide reliable direction for grid resource allocation, job scheduling as well as dynamic load balancing;
- to help grid users to finish computing tasks while minimizing cost on time, space, and money.

2. Representative Systems

2.1. Monitoring Systems

At present, there are a lot of monitoring tools that have been widely used. These tools are usually designed for monitoring individual personal computer or single cluster, so that they can't be used in grid systems directly. However, the monitoring technologies employed and the resource sensors realized are reusable for grid systems. Moreover, these tools are being evolved to support grid system,

* This research work is supported by National Natural Science Foundation of China (60873235 & 61073009) and by Science-Technology Development Key Project of Jilin Province of China (20080318) and by Program of New Century Excellent Talents in University of China (NCET-06-0300). Corresponding author: Xilong Che

along with the rapid development and widespread application of grid computing. In addition, many grid projects have designed the monitoring module of their own. Several representative monitoring systems are introduced as follows:

Hawkeye project [2] can acquire the basic information of host such as host name, identification, processor information, memory capacity, name and version of operating system, file system data, host load, and other basic Condor host data through carrying on the monitoring to the Condor resource pool.

Ganglia project [3] aims at monitoring cluster information including host name, identification, memory capacity, name and version of operating system, file system data, the processor load, and so on through carrying on the monitoring to the cluster.

PAPI project [4] defines a standard API for visiting microprocessor's counter for hardware performance. These counters are like a group of small registers for counting events, when certain signal or relevant state of processor is happening. Monitoring these events can support the correlation between the structure of source/object code and the efficiency of code mapping on architecture, and such correlation are useful for performance analysis and adjustment.

GMA (Grid Monitoring Architecture) project [5] is developed by NASA. It is composed of three parts: sensor, actuator and event service. The exterior sensor calls the Unix utility program (such as df, ps, ping, vmstat, and netstat) to obtain assigned measurement; and the internal sensor can collect the resources usage information through invoking the process. The actuator achieves configuration, process control or other user-customized mission by using command interpreter. The typical functions of actuator are killing processes, sending mail, executing shell command, LDAP service, and so on. The event service maintains tables of publisher and subscriber. It allows the event data transmission from customer process to server process. It also provides monitoring and fault diagnosis for applications, such as host availability, disk space availability, process status of application, and so on.

HBM (Globus Heartbeat Monitor) project [6] is able to monitor Globus system process and the application process at the same time. Heartbeat Monitor consists of three components: client library, local monitor and data collector. The client library supplies APIs for applications to call, and it can also run as an independent procedure. The local monitor checks the client process status periodically, then it renews the local status information and sends a report of process state to the exterior data collector agent. The data collector receives reports from local monitors, then stores these reports in local repository.

2.2. Prediction Systems

Resource monitoring can only provide the instantaneous information of grid node, however, it can not afford to generalize the dynamic variation principle of resources. Such gap can be filled by resource prediction. Lots of Grid middle-wares are born with prediction component, while many efforts are dedicated on integrating prediction tools with projects without such component. Several representative prediction efforts are introduced as follows:

RPS (Resource Prediction System) project [7,8] is a resources-oriented system for online prediction and scheduling. It carries on explicit prediction based on the resource signal, and realizes time series models to predict resource information of hosts. RPS is consist of sensor library, time series prediction library, mirror communication template library, scripts and other auxiliary codes. The sensor library provides acquisition mechanism of resources information to monitoring component, and the time series prediction library provides a scalable, object-oriented C++ template, as well as several linear models for prediction component. It fits data on models and generates prediction through the most appropriate model, then evaluates its performance in application.

NWS (Network Weather Service) project [9,10] is a distributed system for generation and publication of computing resources prediction, periodically and dynamically. It maintains a group of distributed performance sensors, such as CPU sensors, network sensors, etc. NWS collects information from these sensors on computing nodes, and predicts resource usages in certain time interval ahead, using multiple models such as mean based one, median based one, and autoregressive method.

NWS and RPS are supplement to each other. For example, RPS can use NWS sensors, while NWS can use RPS prediction model. Latest versions of the two systems are both extended to support grid systems.

CORI (Collectors of Resource Information) project [11] designs a performance subsystem to enable DIET (Distributed Interactive Engineering Toolbox) project [12] interfacing with third-party performance prediction tools. They also mentioned the importance of prediction, though they didn't propose any prediction method themselves.

GHS (Grid Harvest Service) project [13] is a performance evaluation and task scheduling system for solving large-scale applications in shared environment. Its framework includes predictors in application-level and system-level, as well as interactions with other components within the system. Their efforts are dedicated on the systematic structure rather than prediction methodology, thus their implementation simply uses mean based method, whereas other prediction methods are welcomed to replace theirs, which provides wide extension space for further researches.

3. Monitoring and prediction Strategies

Suppose that a computational Grid contains j nodes, each node has k resource elements rs, which could be host load, bandwidth/latency to certain destination, or available memory usage, etc. $rs \in \mathrm{R}$, $j,k \in \mathrm{N}$, $\mathrm{N}=\{1,2,...,n\}$, then we define Grid resource matrix RS^{Grid} as follows:

$$RS_{j\times k}^{Grid} = \begin{bmatrix} rs_{1,1} & \dots & rs_{1,1} \\ .... & \dots & \dots \\ rs_{j,1} & \dots & rs_{1,1} \end{bmatrix} \tag{1}$$

Each resource element rs can be expressed by $rs(t)$ because its state value varies dynamically. Therefore, the monitoring and prediction on grid resources are realized by the monitoring and prediction on rs, and Grid resource prediction is a kind of regression procedure as far as its essence is concerned [14]. State values of rs are monitored and recorded to form resource time series, denoted as $Z=\{z_u\}_{u=1}^{U}$, where $z_u \in \mathrm{R}$, $U \in \mathrm{N}$. Let z_t stand for value of current time, then $Z_-=\{z_u\}_{u=1}^{t}$, $Z_+=\{z_u\}_{u=t+1}^{U}$ can be used to represent history set and future set separately, where $t \in (1,U)$. We define F: $Z_- \rightarrow Z_+$ as prediction function set, then any element $f \in F$ is a prediction function. In this research, we focus on q-step-ahead prediction function, its definition is given in formula (2).

$$f: z_{t+q}=f(z_t,z_{t-1},z_{t-2},...,z_{t-m+1}),\ \ q,m \in \mathrm{N} \tag{2}$$

The prediction framework is schematically shown in Figure 1. The resource set is divided into three parts: training, validation and test sets. The training set is used to build prediction model, which is optimized using validation set and evaluated using test set. The model takes historical data as input and generates prediction for future variation.

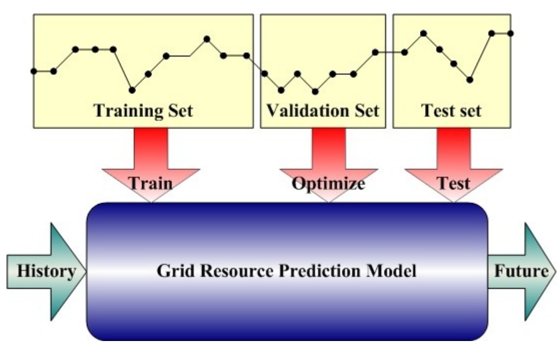


Figure 1. Prediction framework.

3.1. Monitoring Techniques

Monitoring tools are distinguished by resource types realized and monitoring techniques employed. From another point of view, the choices of the monitoring techniques are also decided by the characteristic of resource types. Based on analysis to several representative monitoring systems, monitoring techniques are summarized as follows. Table 1 illustrates the usual resource types and our recommendation techniques for monitoring them.

Table 1. Recommendation techniques for building resource sensors

CPU	Memory	Disk	Load	Network
CPU_usage (c)	Mem_usage (c)	Disk usage (c)	Load1 (c)	Net IO Rate (c)
CPU_sys (c)	Mem_used (c)	Disk used (c)	Load5 (c)	Latency (a)
CPU_usr (c)	Mem_free (c)	Disk free (c)	Load15 (c)	Bandwidth (a)
CPU_frequency (c)	Mem_capacity (c)	Disk capacity (c)	____	____
CPU_Mflops (a)	Mem_IO Rate (a)	Disk IO Rate (a)	____	____

(a) Benchmark Test Sensor

Benchmark test sensor executes certain operations (i.e. I/O operation) and calculates the running performance of resource as monitoring data, such as net latency and bandwidth. It is the most direct method though the information acquired by this method is limited and the efficiency is poor, while some benchmark tools like PAPI also need to recompile the system kernel.

(b) Kernel Module Sensor

Most of operating systems can provide kernel module for monitoring host, and such module can be employed to get monitoring data. It is an effective method for information collection. However, it is hard to maintain code consistency when changes happen in main kernel source. Besides, a kernel module used by user may conflicts with other kernel modules, thus would cause system instability if improperly handled.

(c) Virtual File System Sensor

The virtual file system is a special file system provided in UNIX kind systems. It is in fact an area in memory, although it is stored and handled as file system (usually in /proc directory). The virtual file system is an online reflection of system information, and it is a monitoring method of more safety and efficiency. It can guarantee the data synchronicity and avoid the operations to system kernel module. The variations in kernel are more frequent than in virtual file system, therefore virtual file system sensor will encounter fewer problems in information acquisition.

(d) Mixed Techniques

System information can also be generated by mixed techniques other than individual ones. For example, we can collect data using kernel module, then output such data by way of virtual file system interface.

3.2. Prediction Techniques

Resource prediction is based on resource monitoring. It sums up historical data for modeling, and seeks to find the variation principles of resources, and makes judgment or prediction of short-term or even long-term in future interval. Performance of different models is distinguished by the prediction techniques employed. Several representative ones are discussed and compared as follows:

(A) Linear Time Series

Resource variations are considered a linear time series regression process in many researches [7,8,9,10]. Box-Jenkins models are a series of linear time series ones, which are also well known as AR-class models, including AR (purely autoregressive), MA (purely moving average), ARMA (autoregressive moving average), ARIMA (autoregressive integrated moving average), and ARFIMA (autoregressive fractionally integrated moving average). Subsequently, ARCH (autoregressive conditional heteroskedasticity) and GARCH (generalized autoregressive conditional heterosjedasticity) are also added based on secondary modeling to the error. AR-class models are universal thus other linear models like Markov process or Mean/Median process can also be expressed using AR-class models, partly or completely.

(B) Wavelet Analysis

In this method, resource variations are considered a superposition of multiple waveforms. It generates prediction based on periodicity in variations. It is doing well on signal with periodical behavior, and it has a good self-adaptability. While the drawback is that it is not feasible for the application with too much randomness, therefore it is usually combined with other techniques in modeling, for example in combination with support vector machine [15].

(C) Stochastic Information

It takes resource variations as a stochastic process [16]. This method is based on the assumption that the resource information follows normal distribution. However, it is not the truth in most of practical applications. Its reliability can be improved by adapting the original assumption, that is replacing normal distribution with interval distribution. This method is simple, intuitionistic and fast, while its limitation is that the distribution of the interval values must be unified.

(D) Artificial Neural Network

The ANNs (Artificial Neural Networks) are powerful tools for self-learning, and they can generalize the characteristics of resource variations by proper training. ANNs are born with distributed architecture as well as robustness. They are suitable for multi-information fusion, and are competent for quantitative and qualitative analysis. ANNs have been employed by many researches in resource prediction. In the research of [17], it is indicated that the ANNs prediction are more accurate and outperform the methods in NWS. However, ANN's learning process is quite complex, thus is inefficient in modeling. Furthermore, the choices of model structures and parameters are lack of standard theory, so that it usually suffers from over-fitting or under-fitting with ill chosen parameters.

(E) Support Vector Machine

As a promising solution to nonlinear regression problems, SVM (Support Vector Machine) [18] has recently been winning popularity due to its remarkable characteristics such as good generalization performance, the absence of local minima and sparse representation of the solution. The traditional regression techniques, including neural networks, are based on the ERM (Empirical Risk Minimization) principle, while SVM is proposed based on the SRM (Structural Risk Minimization) principle, which tries to control model complexity as well as the upper bound of generalization risk, rather than minimizing the training error only, thus is expected to achieve better performance than traditional methods. Prem and Raghavan [19] have explored the possibility of applying SVM to forecast resource measures and indicated that the SVMs forecasts are more accurate and outperform the NWS methods, such as Autoregressive ones and Mean/Median based ones.

4. Numerical Simulation

This simulation aims to compare efficiency and accuracy of different models for multi-step-ahead prediction of Grid resources, including BPNN (Back Propagation Neural Network), RBFNN (Radial Basis Function Neural Network), GHNN (General Hybrid Neural Network, which hybridizes RBFNN and BPNN), ESVR (Epsilon-Support Vector Regression), and NSVR (Nu-Support Vector Regression). The model parameters are initialized with values that are commonly used, as is given in Table 2 and Table 3.

Table 2. Parameters for ANNs

parameter	**BPNN**	**RBFNN**	**GHNN**
input layer node number	6	6	6
output layer node number	1	1	1
sigmoid hidden layer node number	5	____	5
RBF hidden layer node number	____	5	5
weight study rate	0.1	0.1	0.1
iteration number	500	500	500

Table 3. Parameters for SVR

parameter	ESVR	NSVR
input feature number	6	6
output feature number	1	1
kernel function	RBF	RBF
radius of kernel	0.025	0.025
regularized constant	1	1
epsilon	0.1	____
nu	____	0.54

4.1. Preparations for experiments

Available bandwidth and host load are two representative resource elements in computing Grid, therefore their benchmark data sets are chosen to evaluate the performance of prediction models. We prefer using public data rather than historical data recorded by ourselves, for the purpose of giving comparable and reproducible results. For available bandwidth prediction, we choose "iepm-bw.bnl.gov.iperf2" [20]. It is published by the Stanford Linear Accelerator Center, University of Stanford. For host load prediction, we choose "mystere10000.dat" [21]. It is published by the Department of Computer Science, University of Chicago. From each data set, we choose the latest 400 spots for experiments. General statistics of resource sets are listed in Table 4.

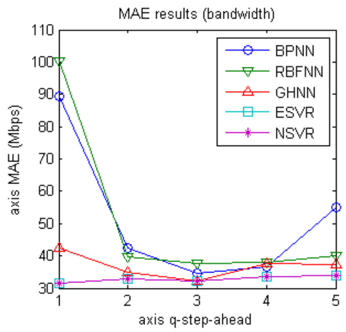
Table 4. Statistics of data sets

General statistic	Bandwidth	Host load
Set size	400	400
Minimum	10.4	0.0
Maximum	335.0	0.61
Mean	82.074	0.113
Variance	3727.851	0.0126

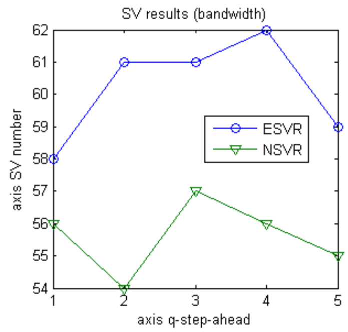
Experimental nodes are running under Fedora Core Linux 9.0 system and connected by 100MB LAN, each node is equipped with single Intel Pentium IV 3.0GHz CPU and 1GB-DDR400Hz memory. We record the training CPU time to measure prediction efficiency, and employ MAE (Mean Absolute Error) to measure prediction accuracy, as in formula (3), where z and z^* denote true value and predicted value in original interval.

$$MAE = \frac{1}{l}\sum_{i=1}^{l} | z - z^* | \tag{3}$$

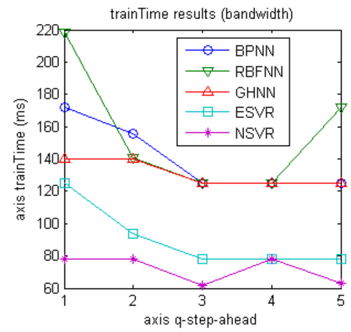
4.2. Results and discussions



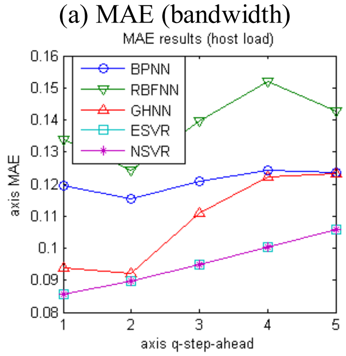
(a) MAE (bandwidth)



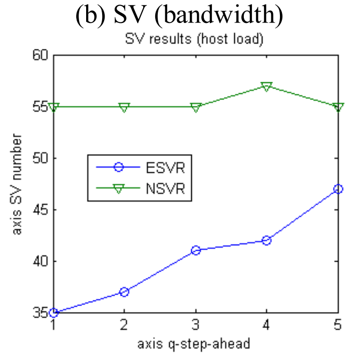
(b) SV (bandwidth)



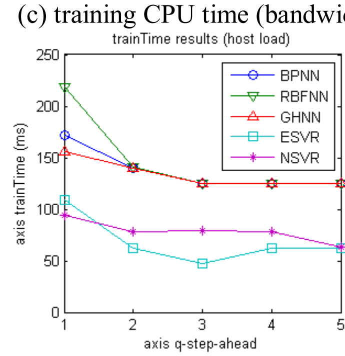
(c) training CPU time (bandwidth)



(d) MAE (host load)



(e) SV (host load)



(f) training CPU time (host load)

Figure 2. Prediction Results.

The MAE results of different models are shown in Figure 2(a) and 2(d). From both figures we can find that GHNN achieves better accuracy than BPNN and RBFNN, while NSVR and ESVR win the best performance in all the q-cases. As prediction step q increases, the prediction error of GHNN, NSVR and ESVR, does not exceed tolerance interval with bandwidth MAE below 40 Mbps and host load MAE below 0.12, which means that these three methods are suitable for resource prediction of both one-step-ahead and multi-step-ahead.

A remarkable characteristic of ESVR/NSVR is the sparse representation of the solution, namely model with less support vectors is better in achieving same accuracy. We can see from Figure 2(b) and 2(e) that the comparison results between the two are data set dependent, in this case we can see that these two methods achieve similar accuracy and complexity.

The training CPU time of individual models is compared in Figure 2(c) and 2(f). From each sub-figure, we can see that the training time does not show a remarkable tendency as step q increases. SVRs cost less time than ANNs, namely within 120ms on both data sets.

5. Conclusions

In this paper, monitoring and prediction mechanisms are discussed to address the problems in building online system for grid resource monitoring and prediction of multi-step-ahead. Our efforts start from analysis and evaluation to representative monitoring and prediction systems. Then recommendations are presented for strategy selection, based on summarization and discussion on typical monitoring and prediction strategies. One-step-ahead prediction is extended to multi-step-ahead prediction, which is then modeled with computational intelligence algorithms such as neural network and support vector regression. Available bandwidth and host load are two typical resource elements in computing Grid, therefore their benchmark data sets are chosen to evaluate the performance of prediction models. Simulation results indicate that support vector regression models achieve superior performance on both accuracy and efficiency.

6. References

[1] L. F. Bittencourt, and E. R. M. Madeira, "A performance-oriented adaptive scheduler for dependent tasks on grids", Concurrency and Computation: Practice and Experience, June 2008, 20(9), pp. 1029-1049.
[2] Condor Hawkeye project. Available at : http://www.cs.wisc.edu/condor/hawkeye/ , June 1, 2010.
[3] M. L. Massie, B. N. Chun, and D. E. Culler, "The Ganglia Distributed Monitoring System: Design, Implementation, and Experience", Parallel Computing, July 2004, 30(7), pp. 817-840.
[4] F. Wolf, and B. Mohr, "Hardware-Counter Based Automatic Performance Analysis of Parallel Programs", In Proceedings of the Conference on Parallel Computing (ParCo'03, Dresden, Germany, September 3), 2003, pp. 753-760.
[5] A. Waheed, W. Smith, J. George, and J. C. Yan, "An Infrastructure for Monitoring and Management in Computational Grids", In Proceedings of the 5th Workshop on Languages, Compilers and Run-time Systems for Scalable Computers, March 2000, selected by Lecture Notes In Computer Science, Vol. 1915, pp. 235-245.
[6] The Globus Heartbeat Monitor Specification. Available at: http://www-fp.globus.org/hbm/heartbeat_spec.html , June 1, 2010.
[7] P. A. Dinda, "Design, Implementation, and Performance of an Extensible Toolkit for Resource Prediction in Distributed Systems", IEEE Trans. Parallel Distrib. Syst., 2006, 17(2), pp. 160-173.
[8] P. A. Dinda, and D. R. O'Hallaron, "Host load prediction using linear models", Cluster Computing, 2000, 3(4), pp. 265-280.
[9] D. M. Swany, and R. Wolski, "Multivariate resource performance forecasting in the network weather service", In Proceedings of the 2002 ACM/IEEE conference on Supercomputing Baltimore, Maryland, November 2002, pp. 1-10.
[10] R. Wolski, L. Miller, G. Obertelli, and M. Swany, "Performance Information Services for Computational Grids", In: Resource Management for Grid Computing, J. Nabrzyski, J. Schopf, and J. Weglarz, editors, Kluwer Publishers, Fall, 2003.
[11] E. Caron, A. Chis, F. Desprez, and A. Su, "Design of plug-in schedulers for a GRIDRPC environment", Future Generation Computer Systems, 2008, 24(1), pp. 46-57.
[12] E. Caron, and F. Desprez, "DIET: A scalable toolbox to build network enabled servers on the grid", International Journal of High Performance Computing Applications, 2006, 20(3), pp. 335-352.
[13] M. Wu, and X.-H. Sun, "Grid harvest service: a performance system of grid computing", J. Parallel Distrib. Comput., 2006, 66(10), pp. 1322-1337.
[14] L.-T. Lee, D.-F. Tao, and C. Tsao, "An adaptive scheme for predicting the usage of grid resources", Comput. Electr. Eng. Pergamon Press, Inc., Tarrytown, NY, USA, 2007, 33(1), pp. 1-11.
[15] Q. Huang, N. Xiao, and B. Liu, "Grid Load Forecasting Based on Least Squares Support Vector Machine", Computer Technology and development, 2007, 17(6), pp. 32-35.
[16] B. Vandy, and G. Bruno, "Brokering strategies in computational grids using stochastic prediction models", Parallel Computing, May 2007, 33(4-5), pp. 238-249.
[17] A. Eswaradass, X.-H. Sun, and M. Wu, "A neural network based predictive mechanism for available bandwidth", In Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05), 2005, pp. 33a.
[18] V. N. Vapnik, The Nature of Statistical Learning Theory, 2nd ed, Springer-Verlag, New York, 1999.
[19] H. Prem, and N. R. S. Raghavan, "A Support Vector Machine Based Approach for Forecasting of Network Weather Services", J. Grid Comput., 2006, 4(1), pp. 89-114.
[20] Bandwidth data set. Available at: http://www.slac.stanford.edu/comp/net/iepm-bw.slac.stanford.edu/combinedata/ , June 1, 2010.
[21] Host load data set. Available at: http://cs.uchicago.edu/~lyang/Load/ , June 1, 2010.