A Resource-based Server Performance Control for Grid Computing Systems [*]

Naixue Xiong[1,3], Xavier Défago[1,2], Yanxiang He[3], and Yan Yang[4]

[1] School of Information Science, Japan Advanced Institute of Science and Technology
(JAIST), Japan
`naixue@jaist.ac.jp`
[2] PRESTO, Japan Science and Technology Agency (JST)
`defago@jaist.ac.jp`
[3] The State Key Lab of Software Engineering, Computer School,
Wuhan University, PR China
`yxhe@whu.edu.cn`
[4] Computer School, Wuhan university of science and technology, PR China
`Y.Yang@mail.ccnu.edu.cn`

Abstract. With the rapid advances in Internet and Grid technique, an increasing number of applications will involve computing systems. These applications in turn create an increasing demand for efficient resource management, request handling policies and admission control. In this paper, we propose an efficient admission control algorithm to protect the critical resource of server and improve the performance of the computing system. Stability of CPU utilization is aimed to protect the server from overload and under-load. It is then beneficial to keep a satisfactory response time of requests, high throughput and less potential loss of service. We analyze the stability in detail and present a method for tuning control gains in order to guarantee the system stability. Finally, we perform simulations to evaluate the performance of the proposed algorithm. Simulation results demonstrate that the proposed algorithm stabilizes the utilization of CPU in the computing system if the control gains are appropriately chosen on the basis of system stability. It then achieves satisfactory performance.

1 Introduction

With the rapid advances in Internet Application and Grid technique, the number of servers increased sharply in recent years, an increasing number of server applications will involve a large of computing systems. For every server, computing means consuming its resources, including CPU slot, memory, bandwidth and so on. When one or several kinds of resources are scare, the server will be regarded as overload. Because web server overload can lead to loss of service and even possible damage to some critical resources due to its over-utilization,

---

web servers need to be protected from overload. Therefore, there is a need of efficient admission control to protect these resources and services in the above computing systems, especially during periods of peak server load. A number of admission control schemes have been studied in [1-6]. In these admission control methods, much attention has been paid to the context of performance metrics, such as the request response time, system throughput and efficient serving rate. The controller software process is embedded in Apache Server Linux OS in [7]. However, in [7], the stability of the network control system is not discussed, while this requirement to a control system in engineering is very important. Stability of control systems can protect the server from overload and under-load. It is then beneficial to keep a satisfactory response time of requests, high throughput and less potentially loss of service. The performance of an unstable system can severely and persistently diverge from the desired performance so as to cause system malfunctioning even worse to lead to the whole system break down [2]. Choosing the proper setting for tuning parameters is also very important for guaranteeing the stability of the whole system and achieving satisfactory performance.

In this paper, we propose an effective admission control scheme that utilizes the information about the resource consumption, which is indicated by CPU utilization. We call the approach resource-based admission control. Due to stability is a key requirement in admission control to achieve satisfactory performance, we focus on the stability analysis of the proposed network controller, and give a procedure for tuning the control gains in detail. The approach is shown by simulations to be able to avoid resource over-utilization and server overload and improve the computing system performance. Explicitly, we compare by simulations the system performance under various control schemes, namely the stable and unstable situations.

2   Design of admission controller

In controller design, an analysis of the computing system models leads to a controller that will achieve the service level objects. We report on an analysis of a closed-loop system using a proportional integral plus derivative (PID) control law. The object is to maintain the target CPU utilization. Using stability analyzing method in classical control theory, we are able to design a PID controller that leads the computing system to be stable and thus avoid the system oscillations in response to requests. Such oscillations are undesirable since they increase variability, thereby resulting in a failure to meet the service level objects.

2.1   Notations

Unless otherwise specified, the following notations are pertain to the considered computing system model:

$m(n)$: the number of maximum parallel threads in the nth interval;

$f(n)$:the feedback information on server's utilization in the nth interval;

u: the target CPU utilization ratio of genetic server;
$r(n)$: the number of requests in the nth interval;
k_P: the proportional control gain;
k_I : the integral control gain;
k_D : the Derivative control gain;
N: the duration of time slots.

2.2   Selection of sampling time

The choice of sampling time is a key factor that affects the performance of the admission controller [2]. In the controlled system, the sampling time not only determines the length of time between successive updates of the control inputs, but also the length of time system outputs are averaged over. In this sense, the sampling time is also an averaging interval. A short sampling time enables the controller to reacts to changes in the system quickly but increases measurement overhead. A long sampling time keeps the controller from overreacting to random fluctuations by averaging out the stochastic of the metrics, but will also yield a slow response. In order to balance these competing goals, the minimum sampling time in network control system must be larger than two special kinds of time [8]. One is the sampling time of SNMP (simple network management protocol) agent itself and the other is the Round Trip Time (RTT) between the controller and controlled device. The first one is easy to be found from the manuals of device, and the second one should be measured on Internet.

Some sampling rules about robustness can be supposed as below. If the sampling interval is less than the minimum RTT, the feedback will be regarded as the $(n+1)$th sampling results, because the $(n-1)$th feedback has been received and the $(n+1)$th sampling result is impossible to be retrieved so quickly. If the interval is equal to or more than the average RTT, the feedback will be regarded as the $(n+1)$th sampling results, because we suppose that the nth sampling result has been lost. On the basis of this simple and practical method, we design the PID controller.

2.3   Controller design and stability analysis

The hardware and software in admission closed loop control system are distributed as shown in Fig. 1. The reference is the desired CPU utilization denoted by u. In this approach, the job of the administrator is shifted from directly setting the tuning parameters to supplying the desired utilization value. At the server, an admission control algorithm checks if the required CPU is available and determines if the arriving request can be admitted into the system. The server is able to service maximum number of requests in a round. The request will be admitted into the system if the number of requests currently being served plus one request is not more than the maximum number of requests. Because an inappropriately designed controller can overreact to performance errors and push a real-time system to unstable conditions, stability is a necessary condition for achieving the desired performance reference and is especially of importance.

Since the direct measurements of $m(n)$ and $f(n)$ is not be acquired, we can use the history values of $m(i)$ and $f(i)$ $(i = 1, 2, ..., N)$ to estimate $m(n)$ and $f(n)$. Therefore we can design a PID controller that can be better served to predict the number of maximum parallel threads in the nth interval than a simple proportional controller [7, 8].
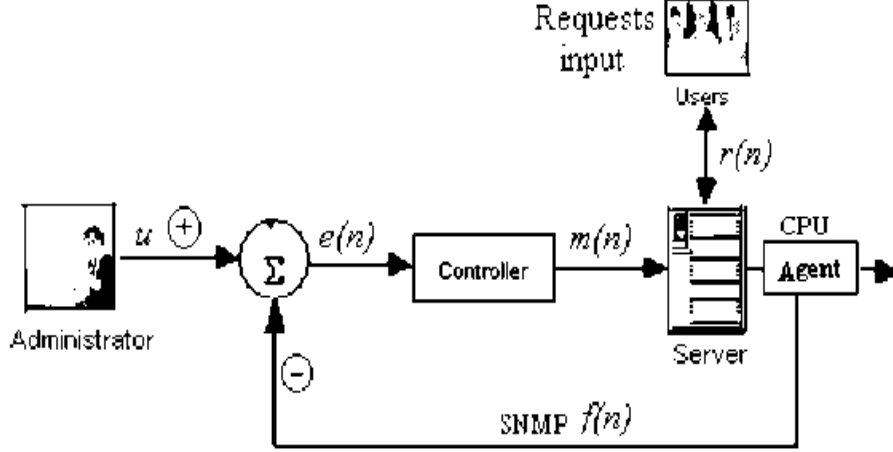

Fig. 1. Block diagram of feedback system for control of CPU utilization

In the following, we propose the PID controller, analyze its system stability and give the specific method for tuning control gains. The PID controller and the state equation of system are respectively described as follows:

$$m(n) = k_P(f(n) - u) + k_I \sum_{j=1}^{N} m(n-j) + k_D[f(n) - f(n-1)]. \qquad (1)$$

$$f(n) = cf(n-1) + am(n-1) + b \sum_{j=2}^{N} m(n-j). \qquad (2)$$

Where a, b and c are constants that are estimated from statistical data. It is noted that the precise description given in (2) can be obtained by using some system identification method on the basis of statistical data. A statistical autoregressive-moving-average (ARMA) is established in [9] to fit the historical measurement. In this method, the procedure of estimating the parameters is as follows: First, measurement of the target system is obtained while configuring the input parameters in a controlled way. Second, the least-squares regression method is used to estimate a,b and c for different values of N. In general, the fit of the model improves as N is increased. For the concrete technique with regard to this modeling process, one is referred to [10].

For analysis purposes, it is much more convenient to convert the above two linear equations from the time domain into the z (frequency) domain, where z is a complex number. z-transfer function has several nice properties. For example, consider two linear systems with transforms $A(z)$ and $B(z)$. Then the transform of the system formed by connecting these two in series is $A(z)B(z)$. If outputs of the two systems are summed, then the combined system has the transform $A(z) + B(z)$. Also, if the input to $A(z)$ is multiplied by k, then the associated transform is $kA(z)$.

Applying these principles to equations (1) and (2), we obtain the z-transform:

$$M(z) = k_P(F(z) - uD(z)) + k_I \sum_{j=1}^{N} M(z)z^{-j} + k_D F(z)(1 - z^{-1}). \qquad (3)$$

$$F(z) = cF(z)z^{-1} + aM(z)z^{-1} + b\sum_{j=2}^{N} M(z)z^{-j}. \qquad (4)$$

Where
$M(z) = \sum_{n=0}^{\infty} m(n)z^{-n}$, $F(z) = \sum_{n=0}^{\infty} f(n)z^{-n}$, $D(z) = \frac{z}{1-z}$.
From (3), we can derive

$$M(z)(1 - k_I \sum_{j=1}^{N} z^{-j}) = k_P(F(z) - uD(z)) + k_D F(z)(1 - z^{-1}). \qquad (5)$$

By substituting (5) into (4), one yields

$$(1 - cz^{-1})(1 - k_I \sum_{j=1}^{N} z^{-j})F(z) = (az^{-1} + b\sum_{j=2}^{N} z^{-j})k_P(F(z) - uD(z))$$
$$+(az^{-1} + b\sum_{j=2}^{N} z^{-j})k_D F(z)(1 - z^{-1}). \qquad (6)$$

From (6), we can get

$$[(1 - cz^{-1})(1 - k_I \sum_{j=1}^{N} z^{-j}) - ak_P z^{-1} - bk_P \sum_{j=2}^{N} z^{-j} - ak_D z^{-1}(1 - z^{-1})$$
$$-bk_D(1 - z^{-1})\sum_{j=2}^{N} z^{-j}] \cdot F(z) = -uk_P D(z)(az^{-1} + b\sum_{j=2}^{N} z^{-j}). \qquad (7)$$

Both sides of (7) are multiplied by z , we can get

$$[(z - c)(1 - k_I \sum_{j=1}^{N} z^{-j}) - ak_P - bk_P \sum_{j=2}^{N} z^{1-j} - ak_D(1 - z^{-1})$$
$$-bk_D(z - 1)\sum_{j=2}^{N} z^{-j}] \cdot F(z) = -uk_P D(z)(a + b\sum_{j=2}^{N} z^{1-j}). \qquad (8)$$

Then the following description $\triangle(z)$ represents the characteristic polynomial of (7)

$$
\begin{aligned}
\triangle z &= (z-c)(1-k_I \sum_{j=1}^{N} z^{-j}) - ak_P - bk_P \sum_{j=2}^{N} z^{1-j} \\
&\quad -ak_D(1-z^{-1}) - bk_D(z-1)\sum_{j=2}^{N} z^{-j} \\
&= z - (k_I + bk_P + bk_D)\sum_{j=2}^{N} z^{1-j} + (ck_I + bk_D)\sum_{j=2}^{N} z^{-j} \\
&\quad -(k_I + c + ak_P + ak_D) + (ak_D + ck_I)z^{-1}.
\end{aligned}
\tag{9}
$$

The above characteristic polynomial is closely related to system stability. From control theory [10, 11], when all the zeros of this polynomial lie within the unit circle, the system governed by (1) and (2) is stable in terms of the server utilization. To yield a condition of stability, we let $k_I = bk_P/(c-1)$, $a = b$ and $k_D = ck_P/(1-c)$. The roots of the equation $\triangle z = 0$ are $z = c$. Therefore, we can get when $|c| < 1$, all roots are in the unit circle. In this case, the controlled computing system is stable in terms of the CPU utilization.

3   Performance Evaluation

In this section, we mainly consider the transient-state response and its stability. Transient-state response represents the responsiveness and efficiency of adaptive resource scheduling in reacting to changes in run-time conditions.

In the following experiments, we investigate if the proposed connection control scheme can be used to protect a web server from overload by a targeted control of high overhead requests. As mentioned in [8, 12], CPU utilization ratio between 60% and 70% is good for it to avoid damage after computing for a long time. Therefore, in our simulations the expected CPU utilization ratio u in simulation is set to be 65%. The values of a, b and c are similar to those suggested in [8, 12], i.e., $a = 0.5$, $b = 0.5$, $c = 0.6$. To simplify the control system, we measure the rule of input requests as "pulse signal" type. The period of such request is two minutes, in the first half period, there're one hundred requests per minute and in the second there are forty. The average input rate is seventy requests per minute.

We perform simulations for different values of a, b, c, k_I, k_P and k_D control gains. The simulation results are shown in Figures 2-5. Obviously, control gains determine the stability of system and the value of N influences the response of system. The smaller N is, the faster the system achieves steady state. When the control gains are chosen in the area of stability (see Figure 2 and Figure 4) based on the above stability condition, the CPU utilization rate is gradually becoming stable after adjusted for a short time. So is the number of maximum parallel threads. On the contrary, when control gains are not chosen in the area

of stability (refer to Figure 3 and Figure 5), the system is unstable, and severe oscillations appear which will result in a failure to meet the service level objects.

The above performance profile establishes a set of metrics of adaptive real-time systems based on the specification of dynamic response in control theory. The metrics enables system designers to apply established control theory techniques to achieve stability, and meet transient and steady state specifications.

4 Conclusions

The widespread use of information technology has motivated the need of performance management of computing systems. In network control system, transfer function is not clear between input and output variables. Therefore, in this paper, we propose the use of a PID feedback control strategy to achieve the goal of optimized performance management. The proposed controller can protect the crucial resource of the server and enhance the performance of the server. Furthermore, systems administrator can translate desired performance into appropriate setting of available tuning parameters. Simulation results demonstrate the proposed algorithm can be applied into the server and can be very effective. Certainly, challenges still exist in the modeling and control of computing systems, for example computing systems generally exhibit stochastic behavior inspired by random requests from users and how to control the performance of them to scale such stochastic behavior would be a subject for future research.
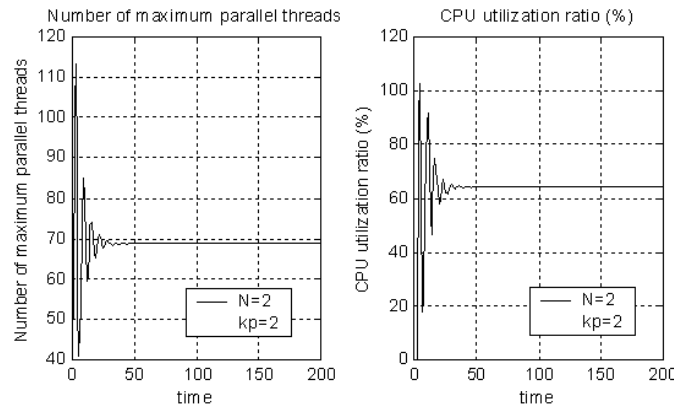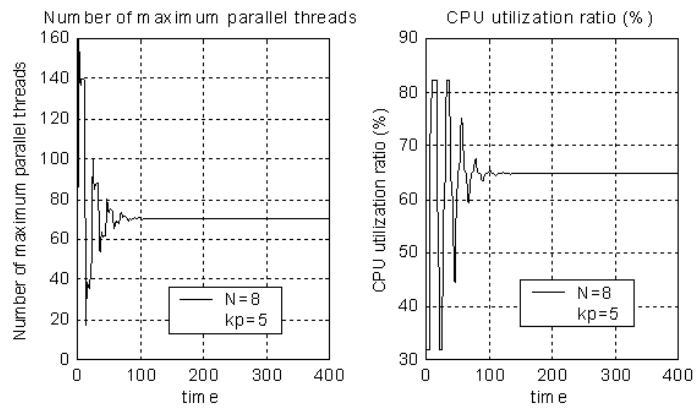

Fig. 2. Control effect of the scheme with parameters $N = 2$, $k_P = 2$ (stable fast responsive case)

References

1. P. Mundur, R. Simon, and A. Sood: Integrated Admission Control in Hierarchical Video-on-Demand Systems. In Proceedings of the IEEE International Conference
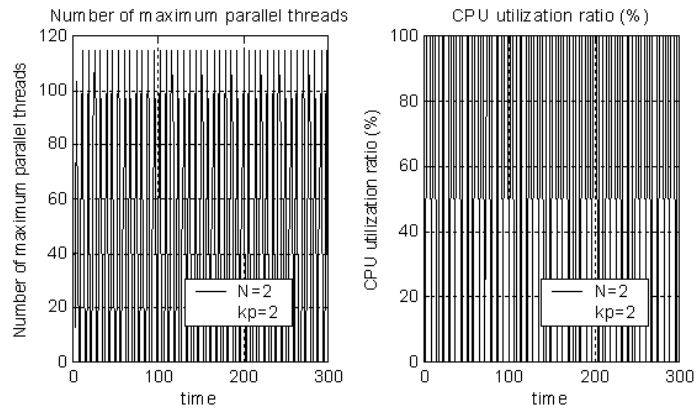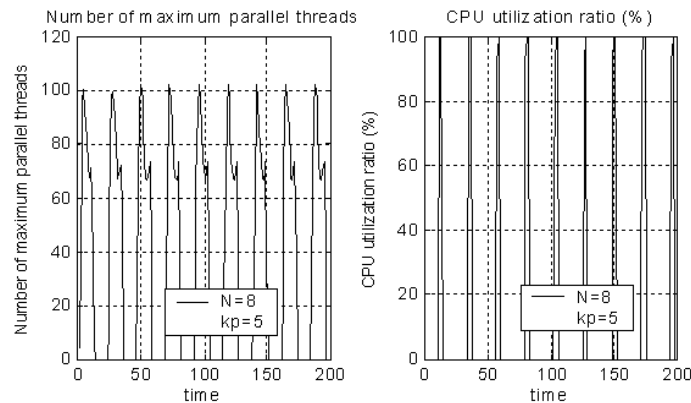
Fig. 3. Control effect of the scheme with parameters $N = 2$, $k_P = 2$ (unstable case)

Fig. 4. Control effect of the scheme with parameters $N = 8$, $k_P = 5$ (Stable but sluggish case)

Fig. 5. Control effect of the scheme with parameters $N = 8$, $k_P = 5$ (unstable case)

on Multimedia Computing and Systems (ICMCS '99), pp. 220-225, Florence, Italy, June 7-11, 1999

2. N. Gandhi and DM Tilbury, Y. Diao, J. Hellerstein, and S. Parekh: MIMO Control of an Apache Web Server: Modeling and Controller Design. Proceedings of American Control Conference, May 2002

3. K. H Yum, E. J Kim, C. R Das, M. Yousif, and J. Duato: Integrated Admission and Congestion Control for QoS Support in Clusters. In Proceedings of IEEE International Conference on Cluster Computing, pp.325-332, September 2002, Chicago, Illinois, pp. 325

4. R. Mortier, I. Pratt, C. Clark, and S. Crosby: Implicit Admission Control. IEEE Journal on Selected Areas in Communications, Vol. 18, No.12, 2000

5. F. Kelly, P. Key, and S. Zachary: Distributed Admission Control. IEEE Journal on Selected Areas in Communications, Vol. 18, No. 12, Dec. 2000

6. R. J Gibbens, F. P Kelly, and P. B Key: A decision-theoretic approach to call admission control in ATM networks. IEEE Journal on Selected Areas of Communications, pp. 1101- 1114, August 1995

7. Thiemo Voigt, Gunningberg: Handling Multiple bottlenecks in web servers using adaptive inbound controls. Seventh International Workshop on Protocols for High-Speed Networks, Berlin, German, April 2002

8. Yijiao Yu, Qin Liu and Liansheng Tan: Application of server performance control with simple network management protocol. The Second International Workshop on Grid and Cooperative Computing, Shanghai, China, December 2003

9. Arnold Neumaier, Tapio Schneider: Estimation of parameters and eigenmodes of multivariate autoregressive models. ACM Transactions on Mathematical Software (TOMS), Volume 27, Issue 1, pp. 27 - 57, March 2001

10. J. L Hellerstein and S Parekh: An introduction to control theory with applications to computer science. ACM SIGMETRICS, 2001

11. W. Kamen, B. S. Heck: Fundamentals of Signals and Systems Using the Web and Matlab. Science Press and Pearson Education North Asia Limited, 2000

12. S. Parekh, N. Gandhi, J. Hellerstein, D. Tilbury, T. Jayram, and J. Bigus: Using control theory to achieve service level objectives in performance management. Journal of Real-time Systems, 23 (1/2), July 2002