Load Balancing in Distributed Systems using Cognitive Behavioral Models

Florin Leon Mihai Horia Zaharia Dan Gâlea

Department of Automatic Control and Computer Engineering
Technical University "Gh. Asachi"
Iaşi

Abstract

Load balancing is a classical problem from the beginning of parallel computing and further in distributed systems design. This is a NP-complete problem. The aim of this paper is to present a new heuristic approach in assuring an optimal load balancing into a local computer network using cognitive behavioral modeling.

Keywords: load balancing, user behavior prediction, neuro-fuzzy modeling and cognitive science

1 Introduction

The problem of load balancing continues to raise interesting challenges to researchers. The operating systems and distributed application design must include solutions for solving it. The main reason is to increase the economic efficiency. The main target of a distributed kernel is to provide supplementary computing power to a local user when is needed. The most efficient solution is to use other partially loaded machines to host computing sub-trees, in other words to use a remote computing solution [5]. This process is known as load balancing and must be automatically made with respect to the end user needs. The classic approach is to use algorithms, which try to solve the problem by analyzing the current state of the system. This way has some disadvantages like the need of medium or large computing power and due to this problem the scalability is poor on larger systems. The other approach is the one, which try to estimate, some how, the future system state in order to propose a more stable solution [1]. This seems to be the future in this domain because the classical approach can drive to dramatic increase of communication into the computer network. In another words the achieved speed-up is very low or worst. This is happening due to the workstation user comportment, which can decide to change his computing needs at any moment. For this reason remote tasks

are unacceptably delayed so from the genitor point of view those can be considered as dead and must be computed locally or sent to other low charged station in the net.

One way of estimating the workstations load is by using a statistical approach such as load functions resulted from repeated measurements. These functions often have a Gaussian aspect, like many other models of natural processes. But these methods ignore, as we previous mentioned, the very cause of the load of a workstation, which is the behavior of the individual user. In the last decades, the progress in cognitive science, especially in cognitive psychology, made possible for the development of behavioral models that can estimate, to a higher or lower degree of precision, the way an individual may act under certain circumstances. While no one can challenge the immense variety of the human nature, it is evident that people tend to act rationally in controlled environments and therefore their behavior is not totally random. This assumption leads to an attempt to discover some general rules that may determine (or at least approximate) the conduct of a person in simple, repeating situations. The combined effect of these rules represents the behavior of the individual.

2 Mathematical Model

We present the mathematical model in a top-down fashion. First, we define the system load:

$$SL = SL(L_1, L_2, ..., L_n) \tag{1}$$

This formula means that the system load depends on the individual loads of the workstations. Usually, the system load is the sum of the individual loads. The load of a workstation depends on several factors, but the most important are the processor activity, the memory use and the fact whether the computer is on or off:

$$L_i(t) = Alive_i(t) \cdot (k \cdot Processor_i(t) + (1 - k) \cdot Memory_i(t)), \tag{2}$$

where $t \in \mathbf{R}$ (to reflect the situation at a certain moment), and $k \in [0, 1]$.

Of course, other indicators may be considered for a detailed analysis: video memory, virtual memory and so on as depicted in the following relation:

$$L = L(I_1, I_2, ..., I_m). \tag{3}$$

These indicators depend on the user behavior, which in its turn depends on a set of generic rules that can be interpreted as action schemata, namely:

$$Beh = Beh(R_1, R_2, ..., R_p). \tag{4}$$

3 Case Study: The SMF Model

Our objective is to predict a pattern for the total load of a distributed system, by modeling the user behavior with three simple rules: the daily working schedule, the user motivation and the user fatigue; hence, the name of the model: SMF (Schedule, Motivation, Fatigue). In this

Table 1: User Behavior Rules Based on Action Schemata.

Schedule	Motivation	Fatigue	Behavior
Off	*Any*	*Any*	Off-duty
On	High	Low	Programming
On	High	High	Internet documentation
On	Low	Low	Game playing
On	Low	High	Internet surfing

case, we adopt a bottom-up approach: starting with these simple rules of individual users, we try to approximate the global behavior of the system.

A daily schedule consists primarily in the start time and the period of working time. These values may vary from a user to another. We considered that $t_{st} = 8 \pm 2$ and $t_w = 8 \pm 1$. Therefore, we defined it as a function $S : \mathbf{R}^3 \to \{0, 1\}$, where S is defined by:

$$S(t, t_{st}, t_w) = \begin{cases} 1 & t \in [t_{st} + 24 \cdot k, t_{st} + t_w + 24 \cdot k] \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

Schedule is an "objective" parameter. However, to define the user behavior, we must also consider "subjective" parameters in the simulation. In order to model the user motivation, we chose a sinusoidal function, with a phase modulation, which depends on the user:

$$M(t, \Delta\varphi) = \frac{1 + sin\left((t + \Delta\varphi) \cdot \frac{2\pi}{T_{MAX}}\right)}{2}, \tag{6}$$

where $M : \mathbf{R}^2 \to [0, 1]$.

The phase modulation can be comparable with a longer period of time, T_{MAX}, for example a week, instead of one day. For simplicity, we wanted all the functions to be normalized between 0 and 1; that is why we added 1 to the sinus and then divided the sum by 2.

Finally, we tried to represent the fatigue, which a user accumulates during a day of work by means of a logistic function. We also defined a critical point, around the half of the working time ($t_w \pm 1$), which signifies the major increase of tiredness a user experience after a certain period of sustained effort.

Thus, a function can be defined, with the property that $F : \mathbf{R}^2 \to [0, 1]$, where a identifies the slope of the function:

$$F(t, t_{fcp}) = \frac{1}{1 + e^{-a(t - t_{fcp})}}. \tag{7}$$

Based on functions (5), (6), and (7), we then can assume some types of user behavior. Because behavior is not a quantitative, but a qualitative concept, fuzzy logic rules and linguistic terms were used in order to define it, as is presented in Table 1.

Table 2: Workstation Load Rules Based on User Behavior.

Behavior	Alive	Processor	Memory
Off-duty	no	-	-
Programming	yes	average	high
Internet documentation	yes	low	low
Game playing	yes	high	average
Internet surfing	yes	low	low

4 Simulation Results

In order to verify our model an application was designed and implemented using Microsoft C# development platform. In our application the user behavior characteristics are modeled using the following data structure:

```
public struct User
{
   public double startTime;
   public double workTime;
   public double motivationDelay;
   public double motivationLimit;
   public double fatigueCriticalPoint;
   public double fatigueLimit;
}
```

While schedule values "on" and "off" can be identified with the crisp values 1 and 0, "high" or "low" motivation or fatigue do not mean 1 or 0 as well. A "low" motivation may be considered if the value of the function previously defined $M < M_{threshold}$ (e.g. $M_{threshold} = 0.33$).

In figure 1, the graphics presented are in accord to the functions (5), (6), and (7), considering a period of five working days.

It must be emphasized that all those values and types of activities are arbitrary, and only studying real users in real working situations can produce a valid ecological model.

Now, with the help of the user behavior, we can approximate the loads of the corresponding workstation. Three load functions were selected as follows:

1. *Alive*, which shows whether the workstation is on or off;
2. *Processor*, which shows the processor activity;
3. *Memory*, which shows the memory use on the workstation.

These functions can be defined in a fuzzy manner, as is presented in Table 2, with the help of the types of behavior already computed:

In figure 2 the effect of using previously mentioned functions in a simulated situation is presented. Here the load of slaves is computed using information from the behavioral model.

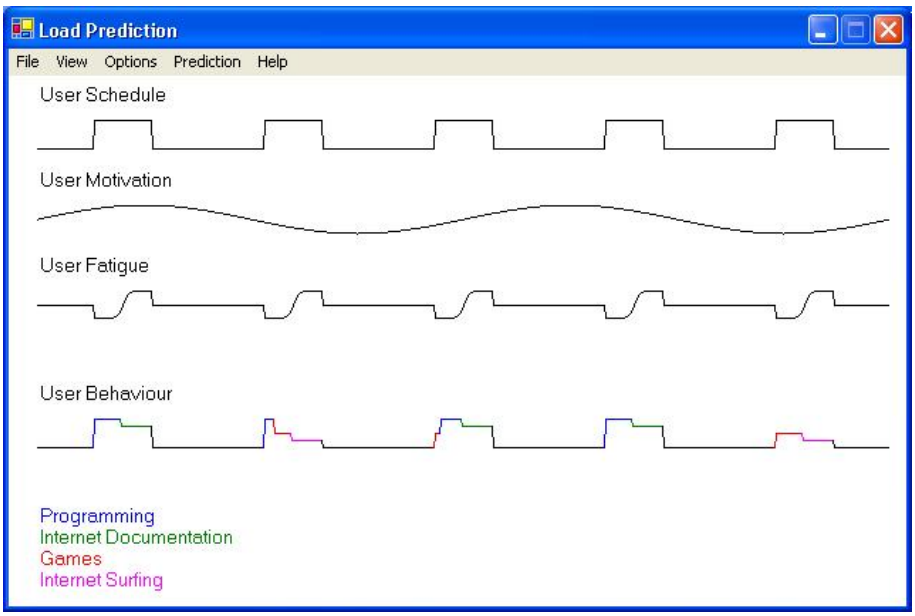

Figure 1: User Behavior in the SMF Model.

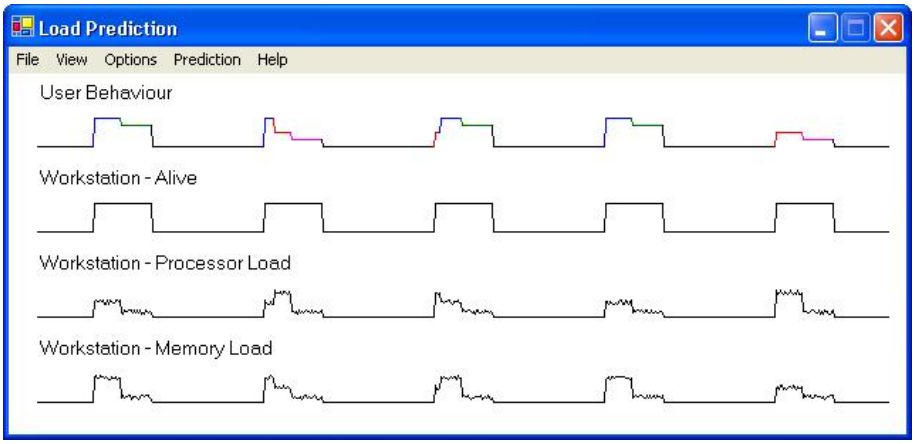

Figure 2: Workstation Load.

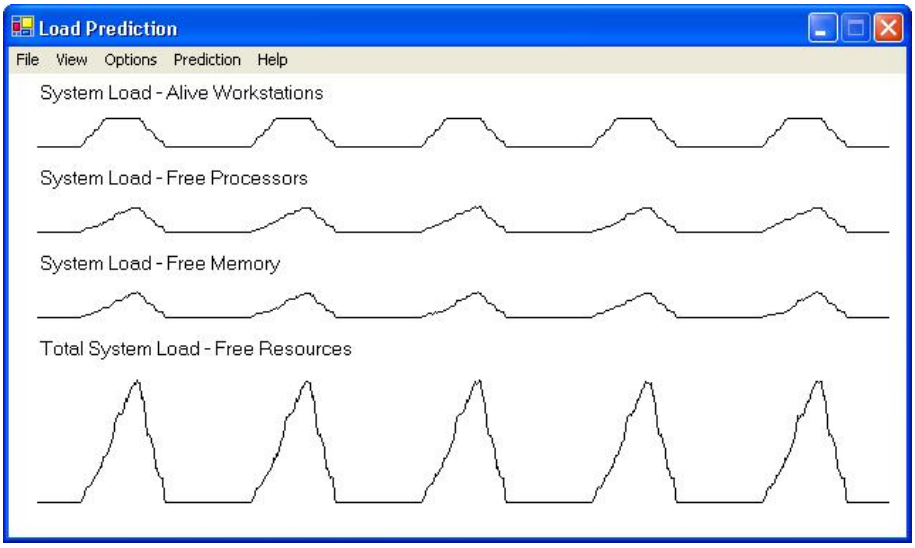

Figure 3: Total System Load, Free Resources.

Having the loads of the individual workstations, we can now compute the overall system load using the following formula:

$$SL = \frac{\sum_{i=1}^{n} L_i}{n} = \frac{\sum_{i=1}^{n} Alive_i \cdot (0.5 \cdot Processor_i + 0.5 \cdot Memory_i)}{n}. \quad (8)$$

Our final step was to predict the evolution of the system load during a day. For this purpose, we used a CANFIS neuro-fuzzy system [3]. It is known that a neuro-fuzzy system is a neural network that is equivalent to a fuzzy inference model. Expert knowledge can be encoded a-priori into its structure, and it can develop, or tune, fuzzy rules [2]. For our example, we used a system with six generalized Bell?shaped membership functions per input and the Takagi-Sugeno fuzzy inference model [4]. Our input was time; we considered prediction for a 24-hour day, as is displayed in figure 3.

After training, we tested the model with different parameters in the simulation. The average error was most of the times lower than 5% as can be seen in figure 4.

5 Conclusions

In the SMF model one can notice that the amount of free resources has the shape similar to the Gaussian function, only slightly deviated towards right.

It is clear that the free resources depend mainly on the number of workstations available. Since the average working schedule is between 8 a.m. and 4 p.m., with certain deviations, this is the period where the workstations are “alive”.

Fatigue is an important factor in the simulation. It is the reason why there are fewer resources available in the first half of the working time, because the users engage in resource-

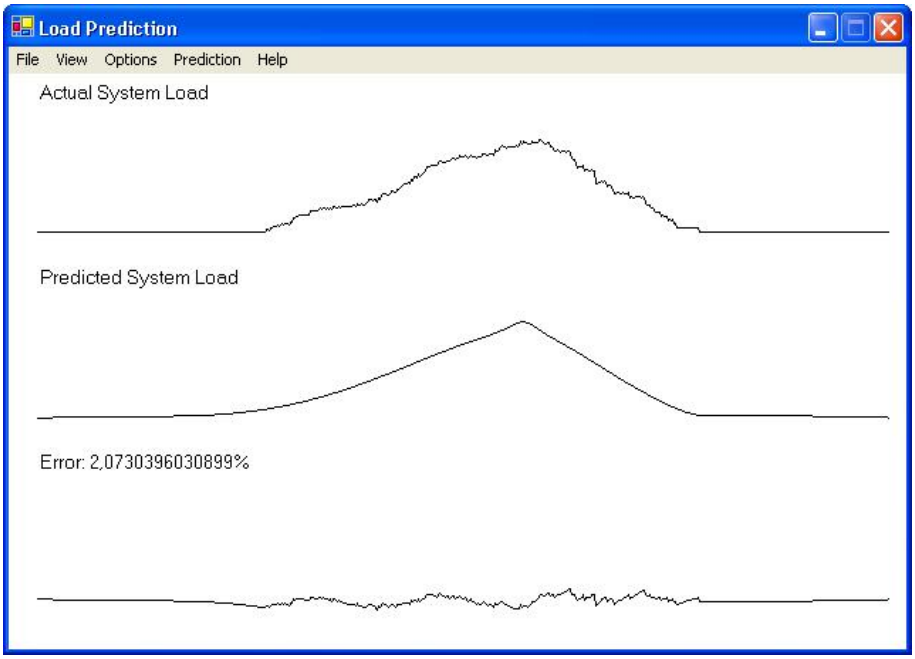

Figure 4: Actual System Load Versus Predicted System Load.

consuming activities. Although each user has different schedules and "subjective" parameters, a statistical effect appears, which allows us to be able to predict the free resources of the system.

The simulation results are in good agreement with the proposed model.

Received, March 3, 2003

References

[1] Hamburg I., Zaharia M.-H., A Strategy for Multimedia-based Distributed Applications in Telecooperation, EURASIP Conf.: ECMCS 2001, Budapest, Hungary, 11-13, 2001.

[2] Hunter A., Parsons S., Applications of Uncertainty Formalisms, 397-415, Springer, Heidelberg, 1998.

[3] Jang J. S. R., Neuro-Fuzzy and Soft Computing, ch. 2, http://neural.cs.nthu.edu.tw/jang/book/slide/ch02.ppt.

[4] Takagi H., Sugeno M., Fuzzy Identification of systems and its Application to Modeling and Control, IEEE Trans on Syst., Man and Cybernetics, 15, 116-132,1985.

[5] Tanenbaum A. S., Distributed Systems, Prentice Hall, NY, 1993.

ECHILIBRAREA ÎNCĂRCĂRII ÎN SISTEMELE DISTRIBUITE FOLOSIND MODELE COGNITIVE COMPORTAMENTALE

(Rezumat)

Se propune o estimare a încărcării dintr-o reţea de calculatoare pe baza unui model comportamental al utilizatorilor. Acest model ales presupune evoluţia a trei factori, de natură externă sau internă: programul de lucru, motivaţia şi oboseala acumulată. Predicţia încărcării se realizează cu ajutorul unei reţele neuro-fuzzy de tip CANFIS.