Volume 2, No. 1, Jan-Feb 2011 ISSN No. 0976-5697
International Journal of Advanced Research in Computer Science
RESEARCH PAPER
Available Online at www.ijarcs.info

Intelligent Fuzzy System for Prediction of Next CPU-Burst Time of Shortest Process Next

Arabinda Nanda*
Department of CSE,
Krupajal Engineering College,
Bhubaneswar, Orissa, INDIA.
aru.nanda@rediffmail.com

Sujit Kumar Panda
Department of CSE,
Krupajal Engineering College,
Bhubaneswar, Orissa, INDIA.
sulinpanda@yahoo.co.in

Sonali Pradhan
Department of CSE,
Krupajal Engineering College,
Bhubaneswar, Orissa, INDIA.
pradhan_sonali@yahoo.co.in

Abstract: SJF scheduling algorithm (SPN or Shortest Process Next), which is one of the best scheduling algorithms among all scheduling algorithm know so far. That always has some problems because when an OS(operating system) wants to execute a process, it does not know the exact execution time of it. After running the process, the exact execution time of that process would come into picture. As a result, for calculating and estimating the execution time of a process before running, many algorithms were introduced. In this paper, intelligent fuzzy system using mamdani fuzzy inference is introduced. This system is able to estimate the CPU brust time prediction degree.

Keywords: Scheduling algorithm, Operating system, CPU-burst time, Fuzzy Inference system, Shortest Process Next (SPN).

I. INTRODUCTION

The main objectives of each scheduling algorithm are as follows: maximum utilization of the processor, maximum throughput, minimum response time or minimum turnaround time and minimum waiting time. Some of the scheduling algorithms are as follows:

FCFS (First Come First Serve) algorithm: In this algorithm, the process which comes first will serve first and does not release it until its processing terminates.

RR (Round Robin) algorithm: In this algorithm, the processes are put in a queue with respect to the arrival time, and take the processor respectively for a slotted time, not until the process is complete. If the processing does not finish in that limited time, the process will be withdrawn (non-preemptive) and be allocated to the next process, and that process is moved to the end of the queue.

SJF (Shortest Job First) algorithm: In this algorithm, the processes are put in the order of their execution time, and the process with the minimum execution time is put at the beginning of the queue. Then the processor is allocated to the processes in respect of the queue order. The SJF algorithm could be designed preemptive or non-preemptive. In preemptive SJF, the process that takes the processor does not release it until it terminates. However, in non-preemptive SJF if a process enters with a less execution time than running process, the processor will be allocated to the new one.

There are so many scheduling algorithms available, each with their own advantages and disadvantages. In this paper, we will focus on SPN algorithm because of minimum waiting time and response time. Because smaller processes are processed earlier, so, bigger processes wait less time and the average waiting time and also response time will decrease. Running a SJF algorithm in a real computer system has some difficulties. When the process comes in the system, its execution time is undetermined. So, how could it be scheduled in a real computer system [2].

First of all, we will present time series concepts and an algorithm used to forecast the processes execution time which acts statistically. Then we will explain fuzzy systems then, we will propose a fuzzy system to forecast the execution time (next CPU-burst time) of a process. Finally, the paper concludes with section V.

II. TIME SERIES AND EXPONENTIAL AVERAGE ALGORITHM

A program in execution state is called a process, and a program is a set of instructions combined together to reach a specific goal. The part of the process which needs a processor in order to be run is called CPU-burst and the part of the process which does not need a processor is called I/O-Burst . A process will run for a while (the CPU-burst time), perform some I/O (the I/O-burst time), then run for a while more (the next CPU- burst time).

A time series is a set of observations which are organized according to the time. Time series prediction is the main objective of this paper. Forecasting the next CPU-burst time of a process which is a kind of time series, will be done by fuzzy inference system. This is done by using the history of past CPU-bursts time which were executed in the processor before. Equation (1) shows the exponential average of the next CPU-burst time:

$$\tau_{n+1} = \alpha . t_n + (1-\alpha) x_n \qquad (1)$$

where

$t_n = Time\ of\ the\ n^{th}\ CPU\ burst\ time$

$\alpha = Forgetting\ co-efficient\ on\ the\ range\ of\ [0,1].$

$\tau_{n+1} = Next\ CPU - burst\ time.$

$Equation\ (1)\ is\ a\ recursive\ equation;\ so\ by\ replacing$

$\tau_n, (2)\ will\ be\ obtained\ as\ following:$

$$\tau_{n+1} = \alpha . t_n + (1-\alpha) x_n \qquad (2)$$

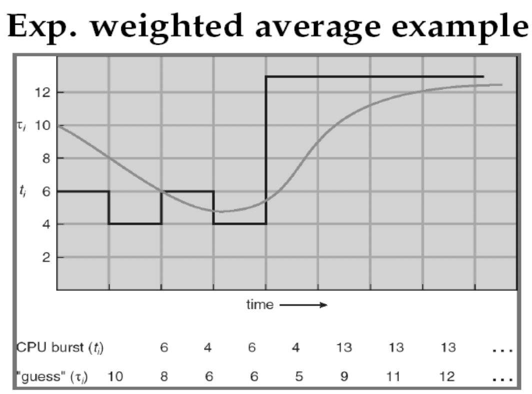

Figure. 1- Forecasted CPU burst time

III. FUZZY CONTROL SYSTEM

Fuzzy system is a knowledge-based rule system [3]. The heart of fuzzy system is a database which was formed with if-then rules. Fuzzy if-then rule is a phrase that some words were marked by membership functions. Membership function (MF) is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) in the range [0, 1]. Fig. 2 shows the main structure of a fuzzy system. In this structure a fuzzy system consists of four sections: Fuzzifier, Inference Engine, Rule Base, and Defuzzifier.

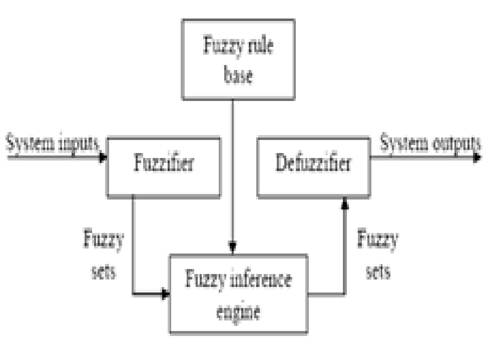

Figure. 2-Structure of Fuzzy Inference system

A. *The Proposed Scheduling Algorithm*

Fuzzy systems are applied for time series prediction. Having known time series and fuzzy systems, we are going to design a fuzzy system to forecast the degree of next CPU-burst time of a process which wants to be scheduled. In this case, the input of the system is the value of the previous used CPU-bursts of a process and the output of the system is the value of the estimated next CPU-burst time. Now, according to the number of the previous observed CPU-burst time, the number of inputs of the fuzzy system should be defined. Therefore, a system pattern extraction is discussed below.

[a] Extracting system patterns

The extraction method of the system patterns is explained with an example. Suppose that a sequence of the past CPU- burst time of a process is shown in the series [t1 , t 2 , t3 ,..., tn].In this series each observation has been showed with t_i which represents the CPU-burst time of the process in the i^{th} observation. Now, for designing a fuzzy system with k inputs, (n-(k+1)) patterns for the system are extracted as follows:

$PAT_1=[t_1,t_2,\ldots,t_k,t_{k+1}]$
$PAT_2=[t_2,t_3,\ldots,t_{k+1},t_{k+2}]$
$PAT_3=[t_3,t_4,\ldots,t_{k+2},t_{k+3}]$
…………………………
…………………………
…………………………
$PAT_{n-(k+1)}=[t_{n-k},t_{n-k+1},\ldots,t_{n-1},t_n]$ (3)

As it seen in, each pattern consists of k+1 element. The first k elements are inputs and the last element is an output. If the fuzzy system can adopt itself with these patterns since according to the each inputs can generate near or the same output of that pattern then, the fuzzy system could be ready for forecasting. Therefore, according to the inputs of (3) the output of the fuzzy system is the predicted next CPU-burst time.

$PAT_3=[t_{n-k+1},t_{n-k+2},\ldots,t_n]$ (4)

[b] Determining of the Number of System Inputs

The determining of the number of inputs in the fuzzy system could have a great influence on its suitable output. The previous experiences have shown that if the number of inputs (patterns) is more, the result of forecasting gets better but slower calculation. There is an important concern that the number of patterns is limited. It was seen in the last section that if the n last CPU-bursts time are exist and the fuzzy system has k input, then the number of patterns would be n-(k+1). Suppose that the aim is to design a fuzzy system with 9 inputs based on the last 15 existing CPU-bursts time. In this case the number of patterns will be 5. Therefore, this system would not have desirable output. Thus the selection of the number of inputs should be done with respect to n. For example, in the above concern, if a system is designed with 3 inputs, a better result would be achieved because the number of patterns was 11.

[c] *Determination of Fuzzy System Membership Function*

Firstly, the number of membership functions for every input should be determined. This could be done based on the previous experiences and in this paper the number of three to five membership functions is suggested for each input. Choosing of many membership functions can increase the complexity of calculations; however, choosing of few ones can decrease the accuracy of calculations. Triangular membership functions are chosen here because of its efficiency and simplicity. If the inputs or the CPU-bursts time were in the range of 0 to 15 ms, then four membership functions for each input would be selected as shown in Fig. 3.

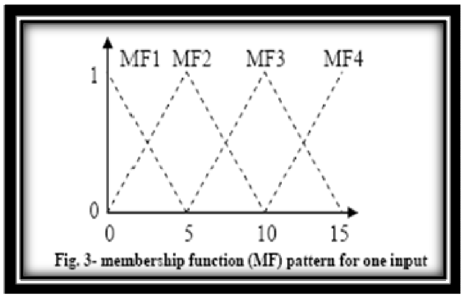

Fig. 3- membership function (MF) pattern for one input

Figure: 3

Selecting of the number of membership functions for the output is more simplified because choosing of many functions does not increase the complexity of calculation but increase the accuracy of calculation. If the number of the output membership functions tends to infinity, the fuzzy system will tend to the Sugeno-type fuzzy system [4]. The number of suggested membership functions to this problem is between five to ten sets.

[d] *Mamdani fuzzy model*

There are 3 types of fuzzy control system/model used.

[i] Mamdani Fuzzy model
[ii] Sugeno Fuzzy model
[iii] Tsukamoto Fuzzy model

In our proposed system, we use mamdani fuzzy model. The most commonly used fuzzy inference technique is the so-called **Mamdani** method [1]. In 1975, Professor Ebrahim Mamdani of London University built one of the first fuzzy systems to control a steam engine and boiler combination. He applied a set of fuzzy rules supplied by experienced human operators.

The Mamdani-style fuzzy inference process is performed in four steps:

[i] Fuzzification of the input variables
[ii] Rule evaluation (inference)
[iii] Aggregation of the rule outputs
[iv] Defuzzification.

Step 1: Fuzzification

The first step is to take the crisp inputs, x1 and y1 (number of input and number of pattern), and determine the degree to which these inputs belong to each of the appropriate fuzzy sets. We examine a simple two-input one-output problem that includes two rules:

<u>Rule: 1</u> IF x is A1 AND y is B2 THEN z is C2

<u>Rule: 2</u> IF x is A2 AND y is B1 THEN z is C1

The Reality for these kinds of rules:

<u>Rule: 1</u> IF number of input is small AND number of pattern is large THEN result is good.

<u>Rule: 2</u> IF number of input is large AND number of pattern is small THEN result is bad.

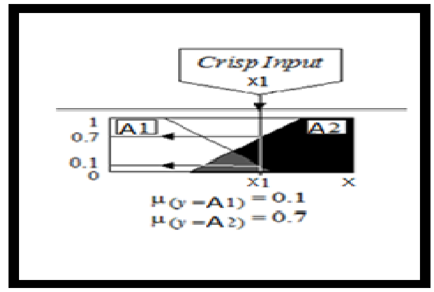

Figure: 4

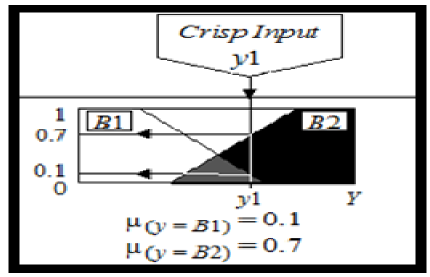

Figure: 5

Step 2: Rule Evaluation

The second step is to take the fuzzified inputs, $\mu_{(x=A1)} = 0.1$, $\mu_{(x=A2)} = 0.7$, $\mu_{(y=B1)} = 0.1$ and $\mu_{(y=B2)} = 0.7$, and apply them to the antecedents of the fuzzy rules. If a given fuzzy rule has multiple antecedents, the fuzzy operator (AND or OR) is used to obtain a single number that represents the result of the antecedent evaluation.

RECALL: To evaluate the disjunction of the rule antecedents, we use the **OR** fuzzy operation. Typically, fuzzy expert systems make use of the classical fuzzy operation union:

$\mu_{A\cup B}(x) = \max\ [\mu_A(x), \mu_B(x)]$

Similarly, in order to evaluate the conjunction of the rule antecedents, we apply the **AND** fuzzy operation intersection:

$\mu_{A\cap B}(x) = \min\ [\mu_A(x), \mu_B(x)]$

<u>Rule: 1</u> IF x is A1 (0.1) AND y is B2 (0.7) THEN z is C2 (0.1)

<u>Rule: 2</u> IF x is A2 (0.7) AND y is B1 (0.1) THEN z is C1 (0.1)

Step 3: Aggregation of the Rule Outputs

Aggregation is the process of unification of the outputs of all rules. We take the membership functions of all rule consequents previously clipped or scaled and combine them into a single fuzzy set. The input of the aggregation process is the list of clipped or scaled consequent membership functions, and the output is one fuzzy set for each output variable.

z is C1 (0.1)→ z is C2 (0.1)=∑

Step 4: Defuzzification

The last step in the fuzzy inference process is defuzzification.Fuzziness helps us to evaluate the rules, but the final output of a fuzzy system has to be a crisp number. The input for the defuzzification process is the aggregate output fuzzy set and the output is a single number. There are several defuzzification methods, but probably the most popular one is the **centroid technique**. It finds the point where a vertical line would slice the aggregate set into two equal masses. Mathematically this **centre of gravity** (**COG**) can be expressed as:

$$COG = \frac{\sum_{x=a}^{b} x.m(x)}{\sum_{x=a}^{b} m(x)}$$

Figure: 6

Centroid defuzzification method finds a point representing the centre of gravity of the aggregated fuzzy set *A*, on the interval [*a, b*].A reasonable estimate can be obtained by calculating it over a sample of points. The output of defuzzification method is the number, will be the degree of predicting next CPU brust time of SJF.

IV. CONCLUSION

By using of intelligent systems such as fuzzy systems, it is possible to estimate a lot of time series including the CPU-burst time series with desirable accuracy. High flexibility and desirable speed in calculation are some advantages of fuzzy systems used in such time series prediction. The main advantage of the proposed system is to forecast the degree of accuracy to produce the execution time of a process (next CPU-burst time) based on past history. Therefore, running a SPN scheduling algorithm or the other algorithms which need to know the next CPU-burst time of processes in a real computer system will come true.

V. REFERENCES

[1] A. Nanda, A. K. Rath, S. K. Rout," Real Time Wireless Sensor Network for Coastal Erosion using Fuzzy inference system": International Journal of Computer Science & Emerging Technology.Vol.-1,Issue-2, August 2010.

[2] A. G pourali,A. M Rahamani," A Fuzzy-based scheduling algorithm for prediction of Next CPU-brust time to implement shortest process Next":International Association of Computer Science and Information Technology"– Spring Conference 2009

[3] Zadeh, L.A., "Fuzzy sets," information and Control, Vol. l. 8, 1965, pp.338-353.

[4] M. Sugeno, "Industrial applications of fuzzy control", Elsevier Science Pub. Co., 1995.

