Journal of Computer Science 7 (8): 1146-1151, 2011
ISSN 1549-3636

A Neuro Fuzzy Technique for Process Grain Scheduling of Parallel Jobs

[1]S.V.Sudha and [2]K.Thanushkodi
[1]Department of Information Technology,
Kalaignar Karunanidhi Institute of Technology, Anna University of Technology,
Coimbatore, 641 402, India
[2]Akshaya College of Engineering, Anna University of Technology, Coimbatore, India

Abstract: Problem statement: We present development of neural network based fuzzy inference system for scheduling of parallel Jobs with the help of a real life workload data. The performance evaluation of a parallel system mainly depends on how the processes are co scheduled? Various co scheduling techniques available are First Come First Served, Gang Scheduling, Flexible Co Scheduling and Agile Algorithm **Approach:** In order to use a wide range of objective functions, we used a rule bases scheduling strategy. The rule system depends on scheduling results of the agile algorithm and classifies all possible scheduling states and assigns an appropriate scheduling strategy based on actual state. The rule bases were developed with the help of a real workload data. **Results:** With the help of rule base results, scheduling was done again, which is compared with the first come first served, gang scheduling, flexible co scheduling and agile algorithm. The results of scheduling showed the optimized results of agile algorithm with the help of neuro fuzzy optimization technique. **Conclusion:** The study confirmed that the Neuro Fuzzy Technique can be used as a better optimization tool for optimizing any scheduling algorithm, This optimization tool is used for agile algorithm which is further used for process grain scheduling of parallel jobs.

Key words: Parallel system, agile algorithm, neuro fuzzy optimization technique, local information, parallel jobs, mean utilization, neural network, fuzzy system, parameter values

INTRODUCTION

Scheduling parallel jobs for execution needs a certain number of processors for a certain time and the schedule have to pack the jobs together. In job scheduling, synchronization overhead could turn to be the key issue for the utilization of the processors. If scheduling does not carefully address the synchronization overhead, the utilization of each processor in a parallel system can end up comparatively lower than a single processor system. Scheduling is done by partitioning the machine's processor and running a job on each partition. Due to the synchronization between processes in job, the jobs do not pack perfectly. If the processes are not co scheduled properly, it will harm the performance of the parallel algorithm. The scheduling algorithm considered is first come first served .gang scheduling, flexible co scheduling and agile algorithm. In the first come first served scheduling, when a job arrives, each of its thread is placed consecutively at the end of the shared queue. When a processor becomes idle, it picks the next ready thread, executes it until it completes or blocks. A set of related threads is scheduled to run on a set of processors at the same time on a one to one basis. The concept of scheduling a set of processes simultaneously on a set of processors uses the threads, which is also called as group scheduling or gang scheduling. Flexible co scheduling is used to improve overall system performance in the presence of heterogeneous hardware or software by using dynamic measurement of applications, communication patterns and classification of application. The algorithm were evaluated with the help of performance metrics like turnaround time ,mean response time, mean reaction time, mean slowdown, average waiting time and mean utilization. The study optimizes the agile algorithm with the help of neuro fuzzy classifier. The rule bases are generated with the help of the scheduling results of the algorithm (Sudha and Thanushkodi, 2008; Jintao, 2010).

Neuro fuzzy system are fuzzy systems that are trained by a learning algorithm derived from neural network theory. The learning procedure operates on local information and causes only local changes to the underlying fuzzy system. The learning process is not

Corresponding Author: S.V. Sudha, Department of Information Technology, Kalaignar Karunanidhi Institute of Technology, Anna University of Technology, Coimbatore, 641 402, India

knowledge based but data driven. A neuro fuzzy system can be viewed as a special three layer feed forward neural network. The first layer represents input variables, the middle layer represents the fuzzy rules and the third layer represents the output variables. The fuzzy sets are encoded as fuzzy connection weights. A neuro fuzzy system can always be interpreted as a system of fuzzy rules. It is possible to create the system out of training data from scratch and it is possible to initialize it by prior knowledge in form of fuzzy rules (Ghedjati, 2010).

The back propagation based neural network, a supervised multilayer feed forward neural network is being used, which accepts the inputs ,process them, producing an output, comparing this output with the desires output and adjusting the weights to produce the better output. Thus the process of learning minimizes the differences between the networks output and the rule base for each pattern in the training set, a rule which best classifies it. The agile algorithm concentrates on the detailed classification of the frequency of synchronization between processes in a system. The processes are classified as fine grain, medium grain, coarse grain and Independent grain workloads.

Fuzzy systems: Within this study, we aim to generate rule based scheduling system. The study concentrates on defining strict boundaries for all the features used as scheduling metrics and the rule assigns an appropriate scheduling algorithm. The study shows the optimized results from a neuro fuzzy system. The assignment of the corresponding scheduling strategy is done only at the end of the scheduling a whole workload trace. The generation of an appropriate situation classification is to be generated during the generation of the rule based scheduling system (Moratori, 2010).

Rule based scheduling system: A fuzzy rule based system is composed of a knowledge base that includes the information in the form of IF THEN fuzzy rules. In linguistic fuzzy rule based system, the knowledge base is composed by a Database (DB) and a Rule Base (RB).A database containing the linguistic term sets considered in the linguistic rules and the membership functions defining the semantics of the linguistic labels. A rule base comprised of a collection of linguistic rules that are joined by a rule connective. From the optimization point of view, to find an appropriate fuzzy model is equivalent to code it as parameter structure and then to find the parameter values that give us the optimum for a concrete fitness function. The parameter values need to be adjusted so that the output of the system fits the desires output. By changing the parameter value, we can able to minimize the deviation between the model's output and the desires output.

In our study, all possible scheduling slots are assigned to the situation class that is described using the already introduced feature. A complete rule base RB consists of a set of rules. Each rule contains a conditional and a consequence part. The conditional part describes the conditions for firing the rule using the defined features and the consequence part describes the scheduling state. In order to specify all scheduling states in an appropriate fashion, each rule defines certain partitions of the feature space within the conditional part (Jintao, 2010; Minh, 2010).

MATERIAL AND METHODS

Scheduling strategy based on neuro fuzzy system: The neural networks and fuzzy systems are dynamic, parallel processing systems that estimate input output functions. They estimate a function without any mathematical model and learn from experience with sample data. The strength of neuro fuzzy system involves requirements like interpretability and accuracy. Neuro fuzzy hybridization results in a hybrid intelligent system that synergizes the two techniques by combining the human like reasoning style of fuzzy system with the learning and structure of neural networks. The main strength of neuro fuzzy systems is that they are universal approximations with the ability to interpret the IF THEN rules. A fuzzy system adaptively infers and modifies its fuzzy association's from representative numerical sample. Neural networks can blindly generate and refine fuzzy rules from training data. Our neuro fuzzy system uses the mamdani's fuzzy model. Each feature in the rule is modeled from a Gaussian membership function. The rules used for the optimization are:

- if awt is small and that is small then the Scheduling Algorithm is class A
- if awt is medium and that is medium then the scheduling algorithm is class B
- if awt is large and that is large then the scheduling algorithm is class C
- if awt is v_large and that is v_large then the scheduling algorithm is class D
- if mrt is small and mret is small then the Scheduling Algorithm is class A
- if mrt is medium and mret is medium then the scheduling algorithm is class B
- if mrt is large and mret is large then the scheduling algorithm is class C
- if mrt is v_large and mret is v_large then the scheduling algorithm is class D

- if msl is small and mu is v_large then the scheduling algorithm is class A
- if msl is medium and mu is large then the scheduling algorithm is class B
- if msl is large and mu is medium then the scheduling algorithm is class C
- if msl is v_large and mu is small then the scheduling algorithm is class D

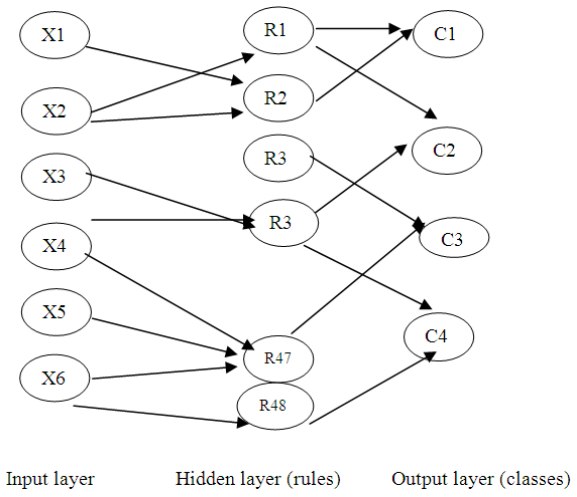
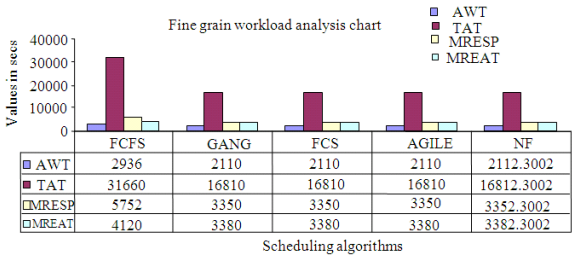
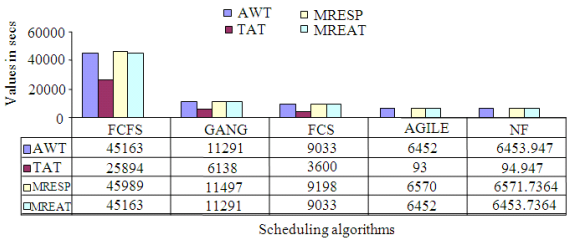


Fig. 1: Learning model for feed forward back propagation Network



	FCFS	GANG	FCS	AGILE	NF
AWT	2936	2110	2110	2110	2112.3002
TAT	31660	16810	16810	16810	16812.3002
MRESP	5752	3350	3350	3350	3352.3002
MREAT	4120	3380	3380	3380	3382.3002

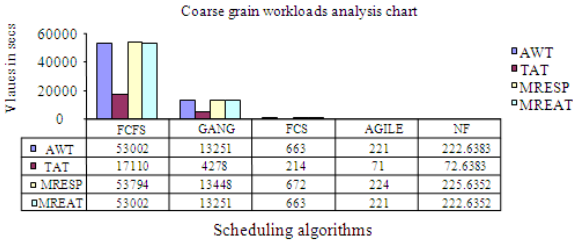
Fig. 2: Fine grain workload-analysis chart



	FCFS	GANG	FCS	AGILE	NF
AWT	45163	11291	9033	6452	6453.947
TAT	25894	6138	3600	93	94.947
MRESP	45989	11497	9198	6570	6571.7364
MREAT	45163	11291	9033	6452	6453.7364

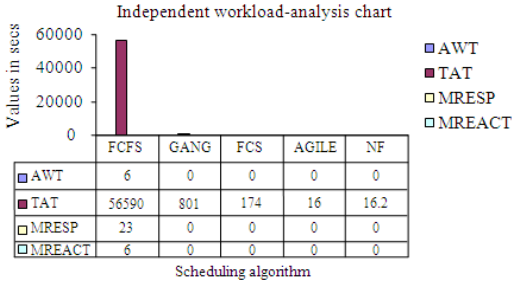
Fig. 3: Medium grain workload-analysis chart

Where awt is the average waiting time, that is the turnaround time, mrt is the mean response time, mret is the mean reaction time, msl is the mean slowdown and mu is the mean utilization. The scheduling classes are Class A is the Agile Algorithm, Class B is the Flexible co scheduling, Class C is the Gang Scheduling and the Class D is the First Come First Serve scheduling. The learning model for feed forward back propagation network is shown in Fig.1The figure shows that there are six inputs in the input layer. The inputs used are average waiting time, mean response time, mean reaction time, mean utilization, mean slowdown and throughput.The hidden layer shows 48 rules used to classify the scheduling classes.The various scheduling classes are Class A is the Agile Algorithm, Class B is the Flexible co scheduling, Class C is the Gang Scheduling and the Class D is the First Come First Serve scheduling (Minh, 2010).



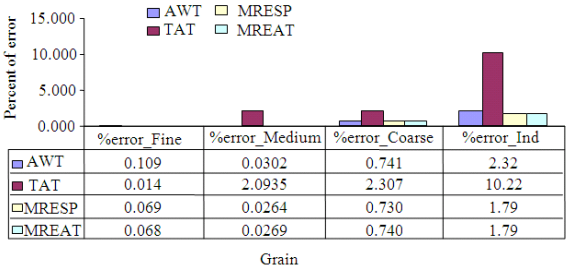
	FCFS	GANG	FCS	AGILE	NF
AWT	53002	13251	663	221	222.6383
TAT	17110	4278	214	71	72.6383
MRESP	53794	13448	672	224	225.6352
MREAT	53002	13251	663	221	222.6352

Fig. 4: Coarse grain workload-analysis chart



	FCFS	GANG	FCS	AGILE	NF
AWT	6	0	0	0	0
TAT	56590	801	174	16	16.2
MRESP	23	0	0	0	0
MREACT	6	0	0	0	0

Fig. 5: Independent grain workload-analysis chart



	%error_Fine	%error_Medium	%error_Coarse	%error_Ind
AWT	0.109	0.0302	0.741	2.32
TAT	0.014	2.0935	2.307	10.22
MRESP	0.069	0.0264	0.730	1.79
MREAT	0.068	0.0269	0.740	1.79

Fig. 6: Error calculation analysis chart

Feed forward back propagation network: Back Propagation neural network is a multilayer feed forward network using a rule based back propagation of error rule. Back propagation provides a computationally efficient method for changing the weights in a feed forward network with differentiable activation function units to learn a training set of input-output pairs. The training algorithm of back propagation involves four stages:

- Initialization of weights
- Feed forward
- Back propagation of errors
- Updating of the weights and trains

During the feed forward stage, each input receives an input signal and transmit this signal to each of the hidden units. Each hidden units then calculates the activation function and sends its signal to each output unit. The output unit calculates the activation function to form the response of the network for the given input pattern (Dutot, 2011).

Algorithm:

Step 1: Initialize the weights to small random values.
Step 2: Perform step 2-4 for each input vector.
Step 3: Set the activation of input unit for xi (I = 1- n)
Step 4: Calculate the net input to hidden unit and its output:

$$Z_{inph} = W_0 + \sum_{j=1}^{6} \sum_{i=1}^{n} (x_{ji} + x_{j+1i}) \quad (1)$$

$$Z_{h} = f(Z_{inph}) \quad (2)$$

Step 5: Now compute the output:

$$Y_{ino} = w_1 + \sum_{h=1}^{6} (Z_h) \quad (3)$$

$$Y_{O} = f(Y_{ino}) \quad (4)$$

Where, x is the input training vector. The various parameters of x are turnaround time, mean response time, mean reaction time and mean slowdown, average waiting time and mean utilization. Z_{inph} is the sum of all the inputs from the input layer. $Z_{h\ is}$ the input to the hidden layer. Y_{ino} is the sum of all the inputs to the output layer from the hidden layer. Y_o is the input to the output layer. w_0 is the bias on the hidden unit. w_1 is the bias on the output unit. The Eq. 1-4 are the activation functions.

RESULTS

The agile algorithm and other scheduling algorithms were executed with the help of real workload data's for the different grains .The implementation is done using Java Programming and the following Table represents the results of the algorithm with the different criteria's. Table 1-4 shows the results of the scheduling job using the algorithms First Come First Served, Gang scheduling, Flexible co scheduling and Agile Algorithm. The Table shows the comparison with the performance metrics like average waiting time, mean response time, turnaround time, mean reaction time, mean utilization and mean slowdown.

The above results show the results of the various grain workloads with the four algorithms and the comparative results are shown with the help of the performance metrics. Using the above results, the neuro fuzzy optimization technique is used, the algorithm is again implemented and the results show that the results of the neuro fuzzy are very close to the agile algorithm. The following Table 5-8 shows the analysis of the comparative results of the FCFS, Gang scheduling, Flexible co scheduling, Agile Algorithm and the result of the optimizations.

The results of the algorithm for the four process grain sizes like fine grain, medium grain, coarse grain and Independent grain were made and is shown in the Figs. 2-5.The Figs. 2-5shows the comparative report of the algorithms like first come first served, gang scheduling, flexible co scheduling, agile algorithm .The above figure also shows the optimized results of the agile algorithm using neuro fuzzy technique. The Fig. 6 shows the error calculations chart of the four process grain sizes.

Table 1: Fine grain workload

Alg/Metric	FCFS	Gang	FCS	Agile
AWT	2936.00000	2110.0	2110.0	2110.0
MRespT	5752.00000	3350.0	3350.0	3350.0
TAT	31660.00000	16810.0	16810.0	16810.0
MReaT	4120.00000	3380.0	380.0	380.0
MeanU	0.50000	0.6	0.6	0.6
MeanS	71.10486	49.2	49.2	49.2

Table 2: Medium grain workload

Alg/Metric	FCFS	Gang	FCS	Agile
AWT	45163.00	11291.00	9033.00	6452.00
MRespT	45989.00	11497.00	9198.00	6570.00
TAT	25894.00	6138.00	3600.00	93.00
MReaT	45163.00	11291.00	9033.00	6452.00
MeanU	0.50	0.50	0.60	0.70
MeanS	1.83	0.46	0.37	0.26

Table 3: Coarse grain workload

Alg/Metric	FCFS	Gang	FCS	Agile
AWT	53002.0	13251.0	663.00	221.000
MRespT	53794.0	13448.0	672.00	224.000
TAT	17110.0	4278.0	214.00	71.000
MReaT	53002.0	13251.0	663.00	221.000
MeanU	0.5	0.6	0.70	0.700
MeanS	3.2	0.8	0.04	0.013

Table 4: Independent grain workload

Alg/metric	FCFS	Gang	FCS	Agile
AWT	6.0000	0.0000	0.0000	0.000000
MRespT	23.0000	0.0000	0.0000	0.000000
TAT	56590.0000	801.0000	174.0000	16.000000
MReaT	6.0000	0.0000	0.0000	0.000000
MeanU	0.5.000	0.5000	0.6000	0.500000
MeanS	3.6912	0.0527	0.0114	0.001036

Table 5: Fine grain workload

Metrics/ algorithms	FCFS	GANG	FCS	AGILE	NF
AWT	2936	2110	2110	2110	2112.3002
TAT	31660	16810	16810	16810	16812.3002
MRESP	5752	3350	3350	3350	3352.3002
MREAT	4120	3380	3380	3380	3382.3002

Table 6: Medium grain workload

Metrics/ algorithms	FCFS	GANG	FCS	AGILE	NF
AWT	45163	11291	9033	6452	6453.9470
TAT	25894	6138	3600	93	94.9470
MRESP	45989	11497	9198	6570	6571.7364
MREAT	45163	11291	9033	6452	6453.7364

Table 7: Coarse grain workload

Metrics/ algorithms	FCFS	GANG	FCS	AGILE	NF
AWT	53002	13251	663	221	222.6383
TAT	17110	4278	214	71	72.6383
MRESP	53794	13448	672	224	225.6352
MREAT	53002	13251	663	221	222.6352

Table 8: Independent grain workload

Metrics/ algorithms	FCFS	GANG	FCS	AGILE	NF
AWT	6	0	0	0	0
TAT	56590	801	174	16	16.2
MRESP	23	0	0	0	0
MREAT	6	0	0	0	0

Table 9: Comparison of error calculations

Metrics/% of error	Error_ fine (%)	Error_ medium %	Error_ coarse (%)	Error_ inde (%)
AWT	0.109	0.0302	0.741	2.32
TAT	0.014	2.0935	2.307	10.22
MRESP	0.069	0.0264	0.730	1.79
MREAT	0.068	0.0269	0.740	1.79

The following Table 9 analysis shows the error calculation of the neuro fuzzy optimization results when compared to the actual scheduling results.

DISCUSSION

The study analyzed parallel job scheduling algorithms in detail and new scheduling algorithm called agile algorithm was discussed (Hangyang., 2011).The agile algorithm was compared with the traditional algorithms like first come first served, gang scheduling, flexible co scheduling with the help of six performance metrics like mean response time, mean reaction time, mean slowdown, turn around time, average waiting time and mean utilization. The study discusses about the optimized technique called neuro fuzzy for the agile algorithm and the results also gives better results for the new algorithm discussed.

CONCLUSION

The Agile Algorithm is optimized using Neuro Fuzzy Optimization. The optimization technique uses the fuzzy inference system, which provides robust inference mechanism with no learning and adaptability and neuro fuzzy algorithm is also superior as it inherits adaptability and learning. From the results it is clearly understood that the agile algorithms are very close to the neuro fuzzy results and thus the agile algorithm is proven to be the better scheduling algorithm for the process grain scheduling of parallel jobs.

REFERENCES

Dutot. P., F. Pascual, K. Zadca and D. Trystram, 2011. Approximation algorithms for the multi-organization scheduling problem. Parallel Distribut. Syst. IEEE Trans., 1-1. DOI: 10.1109/TPDS.2011.47

Ghedjati, F., 2010. Heuristics and a hybrid meta-heuristic for a generalized job-shop scheduling problem. Proceedings of the IEEE Congress on Evolutionary Computing, July, 18-23, IEEE Xplore, Press, Barcelona, pp: 1-8. DOI: 10.1109/CEC.2010.5586004

Jintao, M., Y. Jun and L. Xiaoxu, 2010. Parallel batching scheduling with family jobs for minimizing makespan. Proceedings of the 2nd International Conference on Industrial and Information Systems, July, 10-11, IEEE Explore Press, Dalian, pp: 159-162. DOI: 10.1109/INDUSIS.2010.5565887

Minh. T.N and L. Wolters, 2010. Using using historical data to predict application runtimes on backfilling parallel systems. Proceedings of the 18th Euromicro International Conference on Parallel, Distributed and Network Based Processing, Feb. 17-19, IEEE Xplore Press, Pisa, pp: 246-252. DOI: 10.1109/PDP.2010.18

Moratori. P., S. Petrovic and J.A. Vazquez–Rodringuez, 2010. Fuzzy approaches for robust job shop rescheduling. Proceedings of the IEEE International Conference on Fuzzy systems, July, 18-23, IEEE Xplore Press, Barcelona, pp: 1-7. DOI: 10.1109/FUZZY.2010.5584722

Sudha, S.V. and K. Thanushkodi, 2008. An approach for parallel job scheduling using nimble algorithm. Proceedings of the IEEE International Conference on IEEE Computing and Communicatioin and Networking, Dec. 18-20, IEEE Xplore Press, Thomas, VI., 1-9. DOI: 10.1109/ICCCNET.2008.4787750

Sun, H., Y. Cao, W.J. Hsu, 2011. Efficient adaptive scheduling of multiprocessors with stable parallelism feedback. Parallel Distributed syst., 22: 594-607. DOI: 10.1109/TPDS.2010.121