The Framework for Performance Modeling and Evaluation of Parallel Job Scheduling Algorithms

Amit Chhabra
Assistant Professor, Department of Computer Science & Engineering, Guru Nanak Dev University, Amritsar, India

Gurvinder Singh
Associate Professor, Department of Computer Science & Engineering, Guru Nanak Dev University, Amritsar, India

ABSTRACT
The performance of job scheduling algorithms in campus-wide PC-cluster distributed computing environment may be influenced by several input variables (factors) such as sum of the job sizes of all the jobs in the workload, number of PCs in the cluster and even on the type of scheduling algorithm being used. Response surface methodology (RSM) based statistical regression techniques build empirical model for performance prediction of the scheduling system by means of mathematical equation that relate the scheduler performance (response) to the input process parameters. Artificial neural networks (ANNs) can also be successfully employed for modeling of complex non-linear prediction problems. Feed-forward ANN models viz. multilayer perceptron (MLP) and radial basis functions (RBF) are trained with empirical data to approximate the makespan response of job scheduling algorithms and they can be generalized to predict the new large instances of same problem class. Overall predictive capabilities of these modeling techniques are also measured with various statistical goodness-of-fit standards. This paper will focus on comparing the performance of RSM and ANN based modeling schemes to predict makespan values of job scheduling algorithms in PC-cluster based distributed computing environment. Performance of three space-sharing scheduling algorithms namely First-come-first-serve, Fit-processors-first-served and Largest-job-first is also compared in this paper.

Keywords: PC-cluster, Job scheduling, Response surface methodology, Multilayer perceptron and Radial basis functions

1. INTRODUCTION
Trend of PC-cluster based parallel and distributed computing systems are on the rise in educational institutions due to in- house availability of the required commodity based hardware (i.e. PCs and standard or fast switching network) and software (either proprietary or open source based network operating systems).Space-sharing class of job scheduling algorithms [1,4] tend to distribute the processors of the PC-cluster among competing jobs in a way to produce an efficient schedule. These job scheduling algorithms select the job from the job-queue maintained at job scheduler and allocate the set of PCs to the tasks of selected job. In space-sharing job scheduling algorithms, total processor-space (i.e. machine space) of PC-cluster environment is partitioned into number of processor-partitions where each partition consists of set of processors. Each processor-partition is dedicatedly assigned to the selected job till they are voluntarily released by the job after its completion. Present work focuses on static machine-partitioning approach known as application-based machine partitioning in which number of processors in the partitions cannot change at run-time but partitions may contain different number of processors which is decided on the basis of job's processor requirement information available to the scheduler at the time of arrival of job. Conventional techniques used for evaluating the performance of parallel job scheduling algorithms in distributed computing environments are based on analytical [2, 3, 8] and simulation models [5, 6, 7, 9, 11, 12]. Analytical models can derive mathematical equations relating performance of the system or scheduler to the input parameters that may affect the performance. Though computationally inexpensive but less accurate, they do not always able to characterize the behavior of job scheduling problems in complex diverse distributed systems. On the other hand simulation techniques can model complex systems but are known to be computationally costly and time-consuming. A best possible alternative way is to conduct experiments in a systematic manner to reduce the cost of experimentation and derive empirical-models from the experimental data to explain the mathematical functional relationship between inputs and performance measure of the scheduling process. Application of such empirical techniques viz. statistical models and ANN models for modeling and performance prediction of job scheduling algorithms in parallel and distributed computing environments are not much explored. Apart from providing the mathematical empirical-models, other advantage of statistical modeling techniques over standard one-variable-at-a-time (OVAT) experimental approach is their capability of identifying the main (one variable effect) as well as interaction effect of two variables on the output (performance measure). OVAT approach identify the impact of input factors on the performance metric by varying only one factor and holding all other input factors constant. However OVAT is not capable of estimating the effect on response due to interaction between two variables. Interaction effect is the combined change in two factors that results into an effect greater or less than that of sum of effects expected from either factor alone. Interaction occurs when influence of one factor on response depends on the level of another factor. Application of feed-forward ANN models i.e. MLP and RBF neural network [20] to model and predict the space-shared scheduler performance in large scale PC-cluster environment is another contribution of our present work towards the use of alternative techniques to be used for performance modeling and prediction of scheduling algorithms. Performance of three space-sharing job scheduling algorithms namely First-come-first-serve (FCFS), Fit-processors-first-serve (FPFS) and Largest-job-first (LJF) have been evaluated and compared. This paper uses the Response Surface Methodology (RSM) to conduct systematic and

planned experiments and to build empirical-regression models from the experimental data. Formally RSM [15, 17] is a collection of mathematical-statistical techniques that typically use the classical approach of design of experiments (DOE) [13, 16] for designing planned physical experiments, constructing empirical models, probing the influence of input process parameters (known as factors) on the output variable (known as response) and finding the optimum levels of factors that can result into maximized or minimized response. In this paper, RSM approach has been used to find out the polynomial function approximations to fit the experimental data provided by PC-cluster based experimental (physical) scheduling system. RSM makes use of the multiple linear regression (MLR) analysis to establish the mathematical relationship between process response and the input factors. Multiple linear regression equation helps to establish the influence

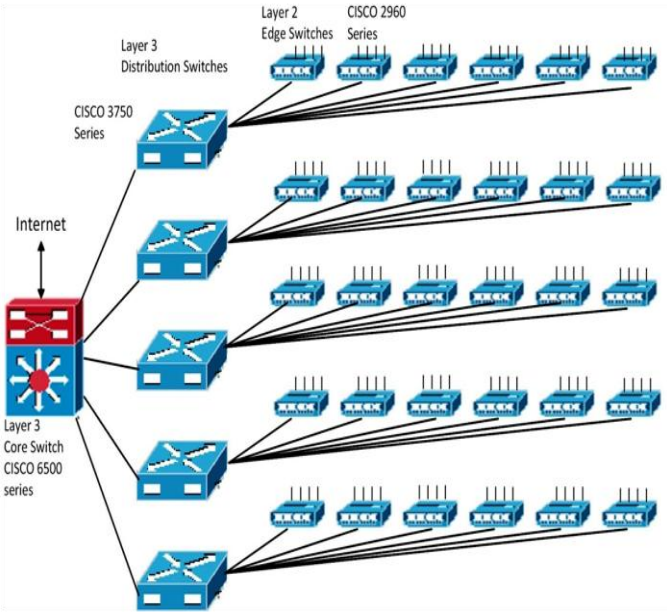

Figure 1: Campus-wide network of PC-cluster

of input factors on the response.

Artificial neural networks (ANNs) with its two most commonly used forms namely feed-forward back-propagation networks (BPN) or multi-layer perceptron (MLP) and radial basis functions (RBF) simulate the parallel computing behavior of neurons of human-brain. Feed-forward ANNs can very well represent the non-linear approximations of pairs of input-output relationships that can be used for solving performance prediction and pattern recognition problems. The generalization and learning capability of ANNs mark them suitable for solving unknown instances of the same problem class once they got trained with the problem data. This paper will focus on using the empirical-statistical modeling techniques viz. RSM, MLP and RBF to predict the makespan response of job scheduling algorithms. Initially RSM approach was applied on the experimental observations to understand the relationship between input factors and response as well as filtering out the non-significant input parameters from the mathematical model. After the RSM phase, MLP and RBF networks will be trained and generalized on the basis of the experimental data containing inputs and output value obtained from the RSM based experimental design phase. Also both the modeling strategies can be used to validate each other's makespan predictions and they are compared with each other using statistical standard metrics.

2. MATERIALS AND METHODS

2.1 Experimental setup and procedure

Campus-wide PC-cluster is constituted with the help of PCs available in the virtual local area networks (VLANs) of various teaching departments of the university campus. These PCs are connected with each other using existing multilayer campus-wide switched network (CWN) shown in figure 1. Currently CWN is in place for providing access to internet and intranet based resources to the hundreds of users in the university. Individual distribution switches are dedicatedly connected to multiple departments. Multiple VLANs are created at these distribution switches to cater group of teaching departments. Distribution switches are further connected to VLAN based access or edge switches which may be further connected to Ethernet switches and hubs to provide end-user access. Server VLANs are formed at the core switch to provide core switch access to distribution switches. Multimode fiber is acting as transmission media between core switch to distribution switches and from distribution switches to edge or access switches. Unshielded twisted pair (UTP) cable is used as a transmission media for connecting edge switches with the user-end switches and PCs. Twenty five PCs are chosen for computation from five VLANs belonging to different distribution switches of the CWN. These nodes are fully dedicated to the PC-cluster however network switches are used in shared mode as they are also used to provide support for internet access and other non-cluster activities to the university users. One of the PCs (with Windows 2003 Server Enterprises edition installed on Pentium core-2-duo, 1GB RAM with gigabit NIC) on the CWN of the PC-cluster is designated as the master PC and the other 24 homogeneous PCs (Pentium IV 3.0 GHz,512 MB RAM and Windows XP) are known as slave or compute PCs (nodes). Master and slave PCs contain the master and slave programs written in java. Master program acts as application-layer abstraction based customized cluster resource management system (CCRMS) to undertake various job scheduling related activities like job submission, job scheduling, job monitoring as well as computing resource allocation and management. Slave program at slave nodes is responsible for communication with the master PC and executing parallel tasks locally. Parallel jobs which typically require exactly the same number of processors as per their processor requirement for their execution are known as rigid parallel jobs. Various data-parallel rigid jobs (Matrix-matrix multiplication, matrix-vector multiplication, pi calculation, image compression and prime no. generation) of varying input sizes in accordance with the square workload model are developed that will act as workload to the job scheduler. In the square workload model every job requests the computing resources (PCs) in the order of n^2 where n is user-defined integer value falling between 1 and 4. Workload will also consist of few sequential jobs. On the basis of their processor requirements, parallel rigid jobs are classified as small and large jobs. User submits the jobs along with their job size or width (i.e. processor requirements) to the job scheduler module residing at master PC using job script file. Master PC with the help of CCRMS based master program schedules and allocates jobs to the processor-partitions consisting of slave nodes with the assistance of its key components viz. job scheduler, job manager and job & status monitor tool. The procedure for job submission, job

scheduling and resource management is shown in figure 2 from step 1 to step 6.

At step 1, user submits all the jobs and their job size information to master node with the help of cluster user interface provided by the master program. All the jobs are held on the job queue maintained by job scheduler module of master program. At step 2, job scheduler module iteratively searches for the jobs in the job queue for scheduling until job queue is empty. At step 3, for example in case of FCFS and LJF scheduling algorithms, a job is selected for execution by the job scheduler only if the available processors match the number of processors required by the job. Otherwise job will have to wait in the job queue till the required number of processors is available due to completion of some other executing

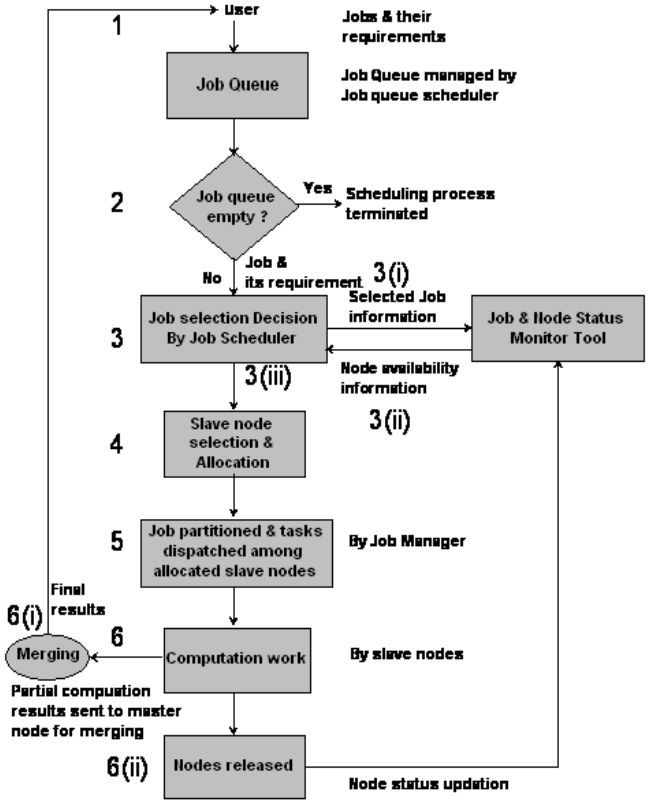

Figure 2: Job scheduling procedure

job. For a similar case in FPFS scheduling, job scheduler scans the jobs in the job queue and picks that job for scheduling for which the sufficient number of processors is available. Therefore FPFS reduces the wait time as compared to FCFS algorithm. Step 4 accounts for selection and allocation of required processors to the job. At step 5, job manager partitions the parallel job into equal-sized tasks depending on the number of processors allocated to the job and dispatches them to the selected set of slave nodes for their parallel execution. At step 6, partial computation results computed by the respective slave nodes are forwarded to merge module of the master program for merging to produce final results. At this particular moment the slave nodes are released and their free status is updated to job & node status monitor tool that in turn updates the status to job scheduler.

2.2 RSM modeling procedure

The RSM approach is used to perform a number of experiments in systematic manner as suggested by experimental design procedure using DOE methodology, based on simulation or physical (actual) experimental data, for a predefined set of design points and to construct a global polynomial approximation of the measured response over the design space. The RSM describes the process response in terms of simple polynomial approximation functions using multiple linear regression techniques.

In CWN based PC-cluster environment, continuous-valued response Y of space-sharing scheduling algorithms can be expressed in terms of explanatory input variables $X_1, X_2, X_3, \ldots, X_n$ using mathematical equation (1). In most of the RSM problems, the true form of this complex relationship $f(.)$ is either unknown or difficult to estimate.

$$Y = f(X_1, X_2, X_3, \ldots, X_n) + e \qquad (1)$$

where e is the statistical error that represents the source of variability due to unaccounted and uncontrolled input parameters in f. This error is considered to be distributed normally with zero mean and variance σ^2.

Using RSM approach the true functional relationship f can be approximated using low-order polynomial approximations with the help of multiple linear regression equation shown in equation (2).

$$Y = g(\boldsymbol{X}, \boldsymbol{\beta}) + e \qquad (2)$$

where $g(\boldsymbol{X})$ is polynomial approximation of function $f(\boldsymbol{X})$, $\boldsymbol{X}$ is a vector containing known quantitative input explanatory or predictor variables $(X_1, X_2, X_3, \ldots, X_n)$ and $\boldsymbol{\beta}$ is a vector composed of unknown parameters also known as regression coefficients of the regression equation whose values can be estimated using method of least squares (MLS). The commonly used polynomial approximation forms for equation (2) can be written as per equations (3)-(5).

First-order model $Y = \beta_0 + \sum_{i=1}^{n} \beta_i X_i + e$ (3)

First-order model with interaction

$Y = \beta_0 + \sum_{i=1}^{n} \beta_i X_i + \sum_{1 \le i \le j}^{n} \beta_{ij} X_i X_j + e$ (4)

Second-order model

$Y = \beta_0 + \sum_{i=1}^{n} \beta_i X_i + \sum_{1 \le i \le j}^{n} \beta_{ij} X_i X_j + \sum_{i=1}^{n} \beta_k X_i^2 + e$ (5)

where n is the number of explanatory variables, β_0, β_i, β_k and β_{ij} are the unknown beta (β) regression coefficients of intercept, first-order, second-order and interaction term respectively. β-regression coefficients determine the relative significance of various input variables in terms of their affect on the response variable. First order polynomial approximation of the function f given by equation (3) is used to represent flat and small response surface containing no curvature. Using equation (3), the response is related to input explanatory variables with the help of linear function. This equation is only used to explain main or first-order effects of explanatory variables. Models represented by equations (4) and (5) are used to characterize the main as well as interaction effects of two or more explanatory variables. These polynomial models are used to represent the curvature in the response surface.

The detailed RSM approach using systematic procedure of DOE for fitting the regression based polynomial approximation model to experimental observations is explained with the help of four phases.

2.2.1 Planning phase

This phase primarily deals with the identification of suitable response (i.e. scheduling performance metric or output) variable and different input process variables (factors) that are supposed to affect the response values in job scheduling process in the CWN based PC-cluster environment. Possible values(qualitative and/or qualitative) or levels i.e. range of input factors are also determined to intentionally notice their measurable impact on the response

variable as the changes in the input factors occurs. The model for the job scheduling system with possible input and output variables is shown in figure 3.

The three input factors chosen in the job scheduling model are sum of the job sizes of all the jobs in the workload known as schedule size (denoted as SchedSize), number of PCs in the cluster called as cluster size (denoted as ClusterSize) and the type of scheduling policy (SchedPolicy) used. These are known as controllable variables because their values can be intentionally varied by the experimenter to see the changes in the response. The input variables are coded or normalized in the range of -1 and +1 to

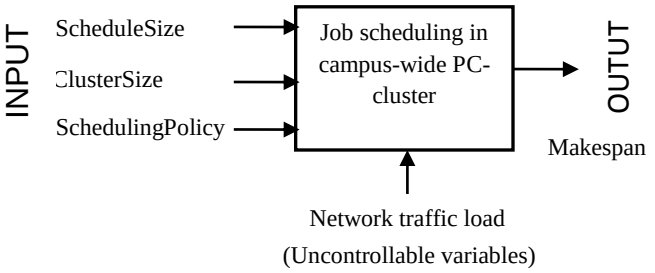

Figure 3: Input and output scheduling process parameters

overcome the effect of biasing of their natural scale and units on the response values. The commonly used method for coding is given below:

$$X_c = \frac{X - (X_{max} + X_{min})/2}{(X_{max} - X_{min})/2}$$

Where X_c is the natural input factor, X is the coded value of natural variable X and X_{max} and X_{min} are maximum and minimum values of the natural variable.

Makespan time is chosen as performance metric or output of the scheduler. It is defined as the total completion time to run a set of jobs and it is an indicator of throughput in offline scheduling systems.

Table 1. Independent process variables and their range

Process variables	Symbols	Levels (actual values)
Schedule Size	SchedSize	73,107,146,177
Cluster Size	ClusterSize	16 -24
Scheduling Policy	SchedPolicy	FCFS,FPFS,LJF

Dynamically varying network traffic load on the various communication sub-systems of the interconnected CWN is considered as uncontrolled or environmental input variable which is difficult to control and characterize during the physical experiment; hence its variability is not accounted by the model. In coarse-grain data-parallel jobs like matrix-multiplication, good amount of communication is involved between master and slave PCs. Communication times between master and slave PCs can change because of the change in network latencies due to varying nature of network load caused by cluster and other non-cluster activities like internet and video-streaming network traffic at different communication switches. Variations in the network traffic load in the non-dedicated kind of multilayered CWN of PC-cluster can cause the presence of random error in the makespan time. This randomness in the response values can be captured by replicating the few experimental readings at the same response surface design points which became the source for measuring the mean square pure error.

2.2.2 Design phase

This phase deals with selection of an appropriate experimental design to determine the number of experiments to be conducted, input variable or factor level combinations for each experimental run and the number of replications of each experimental run. D-optimal experimental design is chosen to collect experimental data and select design points in a way that minimizes the variance associated with the estimates of the specified model coefficients. Design Expert 8.0 trial version software (StatEase Inc. USA) is used to prepare D-optimal experimental design for the current job scheduling problem as shown in table 2.

Table 2. RSM based experimental design for FCFS, FPFS and LJF with experimental and model predictive responses

	Factor 1	Factor 2	Factor 3	Response			
Exp. No.	SchedSize	ClusterSize	SchedPolicy	Makespan(in seconds)			
	Actual (coded)	Actual (coded)	Actual (coded)	Exp.	RSM	MLP	RBF
1	73 (-1.000)	16(-1.000)	FCFS{1 0}	24.039	24.731	24.187	23.829
2	73 (-1.000)	16(-1.000)	FCFS{1 0}	24.785	24.731	24.187	23.829
3	177 (1.000)	16(-1.000)	FCFS{1 0}	58.279	58.856	58.300	58.967
4	177 (1.000)	16(-1.000)	FCFS{1 0}	59.654	58.856	58.300	58.967
5	146 (0.404)	18(-0.410)	FCFS{1 0}	40.501	40.071	40.432	47.127
6	73 (-1.000)	19(-0.283)	FCFS{1 0}	22.872	23.167	22.639	23.829
7	107 (-0.346)	20(0.000)	FCFS{1 0}	27.212	26.457	26.617	23.834
8	177(1.000)	21(0.170)	FCFS{1 0}	49.657	51.387	50.089	48.397
9	107 (-0.346)	23(0.720)	FCFS{1 0}	23.687	23.898	24.589	22.030
10	73 (-1.000)	24(1.000)	FCFS{1 0}	19.839	20.369	20.767	20.570
11	73 (-1.000)	24(1.000)	FCFS{1 0}	20.975	20.369	20.767	20.570
12	177(1.000)	24(1.000)	FCFS{1 0}	46.355	46.088	45.332	48.397
13	177(1.000)	24(1.000)	FCFS{1 0}	47.212	46.088	45.332	48.397
14	73 (-1.000)	16(-1.000)	FPFS{0 1}	22.774	22.200	22.208	25.116
15	107 (-0.346)	16(-1.000)	FPFS{0 1}	27.459	27.481	27.553	25.117
16	177(1.000)	16(-1.000)	FPFS{0 1}	56.130	56.325	55.141	59.999
17	146 (0.404)	18(-0.500)	FPFS{0 1}	37.193	38.002	37.865	36.450
18	73 (-1.000)	20(0.000)	FPFS{0 1}	20.512	20.019	20.748	25.117
19	177(1.000)	20(0.000)	FPFS{0 1}	48.573	49.941	48.975	43.146
20	146 (0.404)	22(0.500)	FPFS{0 1}	32.261	32.870	33.030	36.146
21	73 (-1.000)	24(1.000)	FPFS{0 1}	18.404	17.839	19.732	18.146
22	107 (-0.346)	24(1.000)	FPFS{0 1}	21.384	20.372	22.602	24.146
23	177(1.000)	24(1.000)	FPFS{0 1}	43.441	43.557	43.183	38.146
24	177(1.000)	24(1.000)	FPFS{0 1}	44.031	43.557	43.183	38.146
25	73 (-1.000)	16(-1.000)	LJF{-1 -1}	26.881	27.162	26.365	26.919
26	73 (-1.000)	16(-1.000)	LJF{-1 -1}	27.234	27.162	26.365	26.919
27	177(1.000)	16(-1.000)	LJF{-1 -1}	61.647	61.288	61.644	62.067
28	177(1.000)	16(-1.000)	LJF{-1 -1}	62.487	61.288	61.644	62.067
29	107 (-0.346)	17(-0.720)	LJF{-1 -1}	31.394	31.449	31.874	29.920
30	177(1.000)	19(-0.160)	LJF{-1 -1}	56.291	57.104	56.816	51.462
31	107 (-0.346)	20(0.000)	LJF{-1 -1}	29.763	28.889	28.947	27.080
32	146 (0.404)	22(0.410)	LJF{-1 -1}	36.807	38.295	37.952	41.456
33	73 (-1.000)	24(1.000)	LJF{-1 -1}	21.737	22.801	21.816	21.737
34	73 (-1.000)	24(1.000)	LJF{-1 -1}	22.453	22.801	21.816	21.737
35	177(1.000)	24(1.000)	LJF{-1 -1}	48.231	48.519	48.536	51.462
36	177(1.000)	24(1.000)	LJF{-1 -1}	49.173	48.519	48.536	51.462

D-optimal design utilizes a subset of total possible design points as a basis to solve minimized determinant of fisher information matrix i.e. min $|X^T X|^{-1}$, which minimizes the volume of the confidence ellipsoid for the coefficients resulting into reduction in the total experimental runs required to model the response. A coordinated-exchange point selection algorithm is used to select design points from the potential candidate point set that are spread throughout the experimental design space.

2.2.3. Conduct phase

In this phase experiments are conducted according to the experimental design with the help of experimental setup and procedure described in section. Sometimes few pilot or trial experimental runs have also been conducted to check the consistency of the experimental components and procedure [13]. This probably gives a chance to practice the overall experimental procedure and also gives opportunity to rethink or modify some of decisions taken in earlier phases. Experimental results in terms of makespan metric are recorded and passed to the next phase for statistical analysis.

2.2.4. Analysis phase

This phase deals with analyzing and interpretation of experimental results to derive valid and objective conclusions with the help of DOE based Design Expert 8.0 trial version software (StatEase Inc. USA) [19]. Primarily this phase helps in determining and quantizing the input process or design parameters that are significantly affecting the makespan values. Empirical or predictive model of the experimental data is derived with the help of multiple linear regressions. Regression coefficients are estimated using method of least squares which estimates them in a way so that sum of squares of the errors is minimized. Factorial or n-way ANNOVA is applied on the experimental data to establish the presence of main and interaction effects of input variables. Significance of the predictive model and its model terms can also be judged with the help of ANNOVA analysis. Before conclusions from ANNOVA analysis and regression models are being accepted, adequacy of the fitted model is checked with various graphics based model diagnostic tools to see fitness of the predictive model towards experimental data. Finally few follow-up additional experimental runs were performed to validate the model interpolated predictions.

2.2.4.1 Results and discussion

Actual experimental data is fitted to quadratic regression model supposing the presence of curvature in the response values and to extract main as well as interaction effects present in the model. ANNOVA technique (shown in table 3) is applied on the experimental data to check the response variance in the model due to each input process parameter and their interactions. ANNOVA explains the total variance of the model in terms of individual variance posed by each independent variable in the model. ANNOVA analysis indicated that the model and all other model terms except $ClusterSize^2$, (SchedSize x SchedPolicy) and (ClusterSize x SchedPolicy) are significant at $p<0.0001$. The Model F-value of 1796.86 implies the model is significant.

There is only a 0.01% chance that a "Model F-Value" this large could occur due to noise.Insignificant model terms (with p-value > 0.05) are eliminated from the model equation. Model statistics viz. adjusted coefficient of determination (adjusted-R^2) =0.9968 and insignificant lack of fit (LOF) value 0.1487 indicates the goodness-of-fit of the reduced quadratic model to accurately model and predict makespan values.

Adjusted- R^2 value 0.9968 indicates that 99.68 % of variation around the mean is explained by the model, adjusted for the number of terms in the model and it decreases as the number of terms in the model increases if additional terms don't add value to the model. LOF value represents variation of the data around the fitted model. If a model has a significant LOF value then it is not a good predictor of the response and should not be used [18].Therefore insignificant value of LOF in the present quadratic model represents the fitness of the model to predict makespan values. Major source of variance in the model is due to SchedSize (88.15%) followed by ClusterSize (6.48%) and SchedSize (3.31%). Adequate precision value 121.948, which is a measure of the range in predicted response relative to its associated error i.e. a signal to noise ratio(S/N) indicates an adequate signal for the model to navigate the design space.

Table 3: ANNOVA results of the makespan model

	Makespan (in seconds)				
	Reduced quadratic model				
Source	**Sum of squares**	**df**	**Mean square**	**F-value**	**p-value* (Prob. > F)**
Model	7038.534	6	1173.089	1796.855	< 0.0001
SchedSize	6221.227	1	6221.227	9529.239	< 0.0001
ClusterSize	457.895	1	457.895	701.371	< 0.0001
SchedPolicy	139.170	2	69.585	106.585	< 0.0001
SchedSize x ClusterSize	94.988	1	94.988	145.495	< 0.0001
$SchedSize^2$	234.247	1	234.247	358.803	< 0.0001
Residual	18.933	29	0.653		
Lack of Fit	15.408	20	0.770	1.967	0.1487#
Pure Error	3.525	9	0.392		
Cor. Total	7057.467	35			
Model statistics: S.D: 0.808 C.V. % :2.218 R^2: 0.9973 Adjusted- R^2 : 0.9968 Predicted- R^2 : 0.9960 * Significant at $p\leq0.0001$ #not significant at $p\leq0.05$					

2.2.4.2 Model adequacy checking

Before proceeding towards interpretation of results using empirical model equation of the response i.e. makespan time, it is necessary to check the adequacy of the model using various graphical model diagnostic tools. Normal probability plot of studentized residuals (shown in figure 3) is a graph with a y-axis that is scaled by cumulative probability (Z) that shows at a glimpse whether a particular set of data is normally distributed. The predictive quadratic model passes the normality test as all the design points falls on the straight line. Plot of studentized residuals versus predicted values was diagnosed, which shows the presence of constant variance of the studentized residuals. A layman approach to check constant variance in this plot is to look for absence of outward- opening funnel or megaphone structure of residuals. Presence of outward- opening funnel or megaphone structure indicates that variance of the experimental observations increases as the magnitude of the observations increases [13].

Another diagnostic plot of externally studentized residuals was checked to see the presence of outliers i.e. influential values, however all the residuals falls within the permissible range of ± 3.5. Box-Cox plot was looked for the power transformation suggestions to further improve the model. Power or variance-stabilizing transformations were required in the cases when the max to min ratio of response is greater than 10 i.e. a case representing non-constant variance of the residuals and/or presence of non-normality in the residual data. However no power transformations were suggested in the present case which strengthens the constant variance assumption of the observations.

2.2.4.3 Empirical model building

After conducting the experiment, collecting the experimental response values, performing ANNOVA based statistical analysis and investigating the adequacy of the model using diagnostic plots, the practical conclusion in terms of relationship of the response with the input process variables can be drawn by fitting the empirical-regression model equation to the experimental data. Reduced quadratic model for makespan using coded values of input process variables is derived with the help of multiple linear regression analysis and given in equation 6.This model illustrates the relationship between the makespan and the input explanatory process variables and also can be used to predict the optimized makespan values for various combinations of input process variables at their best levels.

Makespan = 30.8033 + 14.9609 SchedSize – 4.2825 ClusterSize + 0.0331 (SchedPolicy=FCFS) – 2.4978 (SchedPolicy =FPFS) – 2.1019 (SchedSize x ClusterSize) + 6.6746 $SchedSize^2$ (6)

Regression coefficient estimate for the SchedPolicy (LJF) is not by default present in the equation (6) because by default the regression equation can give coefficient estimates for only two levels of qualitative input variable. Regression coefficient for the third level of SchedPolicy variable is determined by calculating the negative sum of all the coefficients of the qualitative variable as given in equation 7.

Coefficient of SchedPolicy (LJF) = – (Coefficient of (SchedPolicy=FCFS)) + (Coefficient of (SchedPolicy=FPFS)) = 2.4647 (7)

Putting coefficient values in equation (6), the final reduced quadratic predictive model for Makespan values is shown in equation (8).

Makespan = 30.8033 + 14.9609 SchedSize – 4.2825 ClusterSize + 0.0331 (SchedPolicy =FCFS) – 2.4978 (SchedPolicy =FPFS) + 2.4647 (SchedPolicy =LJF) – 2.1019 (SchedSize x ClusterSize) + 6.4647 $SchedSize^2$ (8)

Equation (8) shows that most important factor affecting the makespan response regardless of the scheduling policy used is schedule size indicated by SchedSize (with regression coefficient +14.9696) which represents the sum of job sizes of all the jobs in the input workload. As the value of SchedSize increases the makespan values are bound to increase. ClusterSize variable i.e. the number of PCs in the cluster is negatively correlated to the makespan so increase in ClusterSize results into decrease in the makespan value. Regardless of the scheduling policy, interaction term (SchedSize x ClusterSize) shows the antagonistic effect on the makespan response with the regression coefficient of – 2.1019.This shows that combined effect of SchedSize and ClusterSize variables together can result into decrease of makespan response. Terms like (SchedPolicy =LJF) are Boolean expressions whose value is equal to one when the expression is true. Out of the three scheduling policies LJF (with regression coefficient +2.4647) is relatively more affecting the makespan response resulting into larger makespan values followed by FCFS policy (with regression coefficient +0.0331) and FPFS policy which delivers the lowest makespan values due to negative value of regression coefficient (-2.4978) and this trend is also evident from the main effect plot (shown in figure 4) of SchedSize vs. makespan time for all the

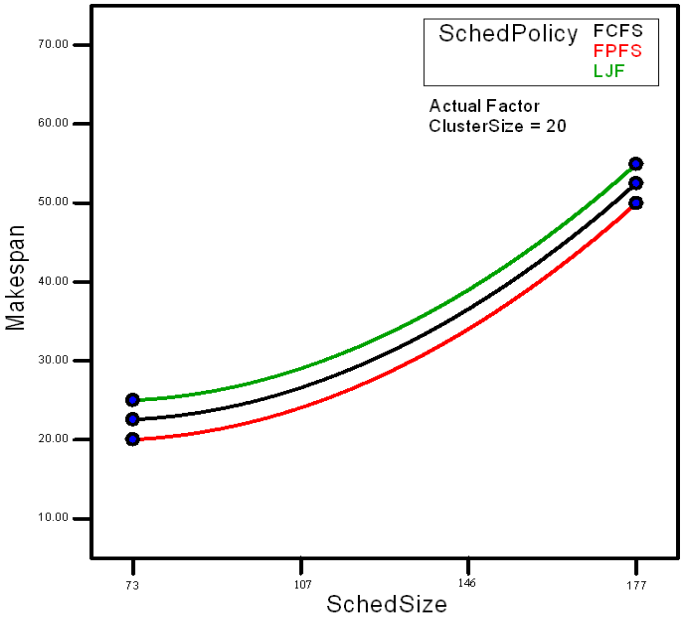

Figure 4: Makespan vs. SchedSize

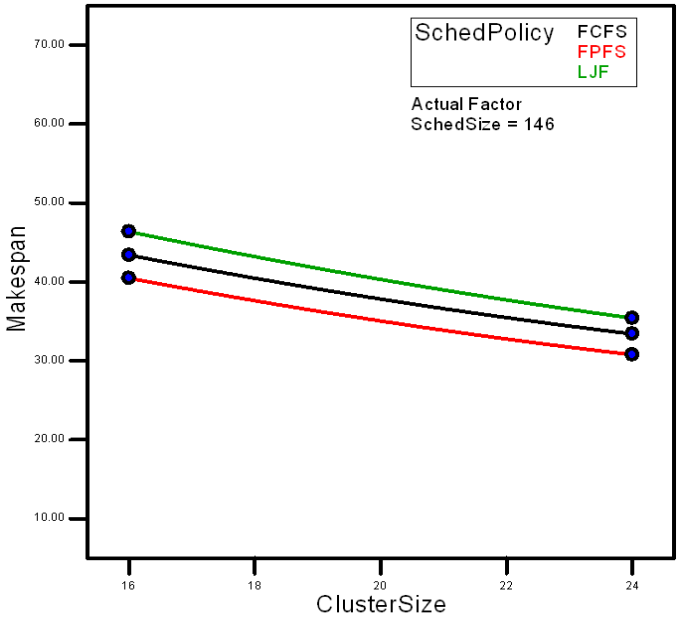

Figure 5: Makespan vs. ClusterSize

three scheduling policies. Regression coefficient (+14.9609) of term SchedSize in equation (8) indicates that a unit change in the value of SchedSize variable can result into 14.9609 amount of change in the makespan value keeping all the input factors constant.

Empirical model equation (8) can be reduced into three individual model equations each for FCFS, FPFS and LJF respectively when their respective Boolean terms are true.

Makespan for FCFS = 30.8364 + 14.9609 SchedSize – 4.2825 ClusterSize – 2.1019 (SchedSize x ClusterSize) + 6.4647 SchedSize^2 (9)

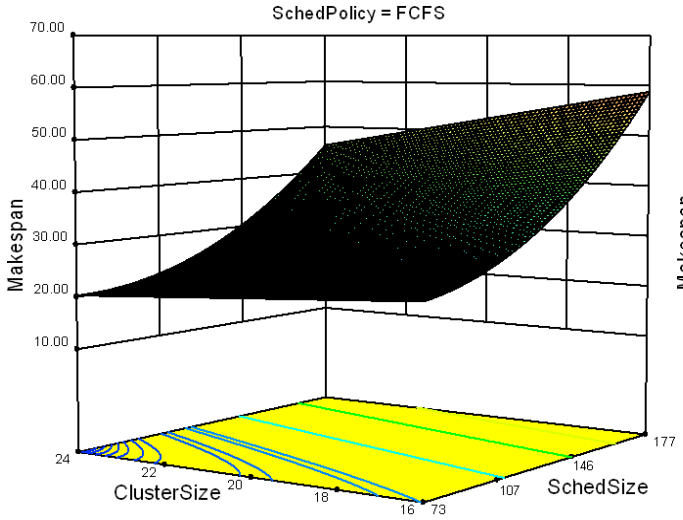

Figure 6: Response surface graph for FCFS policy

Makespan for FPFS = 28.3055 + 14.9609 SchedSize – 4.2825 ClusterSize – 2.1019 (SchedSize x ClusterSize) + 6.4647 SchedSize^2 (10)

Makespan for LJF = 33.2680 + 14.9609 SchedSize – 4.2825 ClusterSize – 2.1019 (SchedSize x ClusterSize) + 6.4647 SchedSize^2 (11)

Looking at equations (9) – (11) for makespan models of FCFS, FPFS and LJF respectively, all the terms have the same regression coefficients with the only difference lies in the intercept values viz. 30.8364, 28.3055 and 33.2680 respectively. Main effect plot of SchedSize vs. makespan time with ClusterSize fixed at 20 is shown in figure 5 which clearly indicates that LJF is producing higher values of makespan (due to higher intercept) followed by FCFS and FPFS policy (giving the lowest makespan time due to lowest intercept value 28.3055).

Similar trend of makespan time has been observed in all the levels of ClusterSize variable for all the scheduling policies leading to a conclusion that LJF produces higher makespan values at all the levels of ClusterSize factor followed by FCFS and FPFS scheduling policies. Main effect plot of makespan time vs. ClusterSize at fixed factor SchedSize=146 indicates the decrease in the makespan values as the cluster size i.e. number of PCs in the cluster increases and this is true for all the three job scheduling algorithms.

2.2.4.4 Optimized makespan time for scheduling policies

Response surface graphs and desirability criteria using Design Expert 8.0 software can be used to find optimized (minimized) makespan values at each of the discrete numeric levels of SchedSize variable. The response surface graphs for the three scheduling algorithms are given in figure x-z respectively. It is evident from the figures 6, 7 and 8 that minimized values of makespan metric at all the numeric discrete levels (i.e. 73,107,146 and 177) of SchedSize factor for FCFS, FPFS and LJF respectively will appear either on or near the left edges of the response surfaces when ClusterSize is at its maximized level i.e. at around 24. It can be judged very well from the response surface graphs in figures 6,7

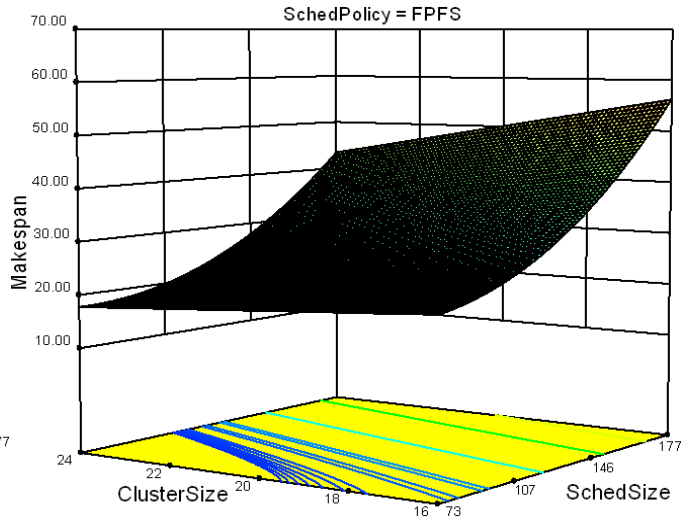

Figure 7: Response surface graph for FPFS policy

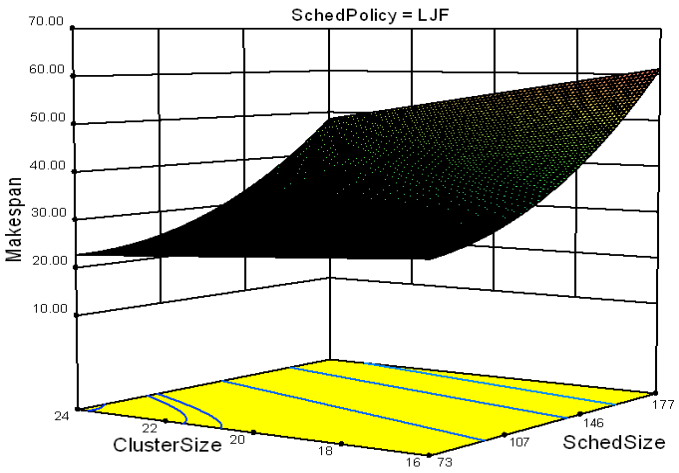

Figure 8: Response surface graph for LJF policy

and 8 that maximized makespan is at the low value of ClusterSize i.e. at 16 and as the cluster size is increased from its low value to high value i.e. 16 to 24, makespan values are greatly decreased. One more observation that can be made from all the three response surfaces that a sharp edge at the makespan time axis is observed when SchedSize is minimum i.e. equal to 73 and ClusterSize is increased from 23 to 24 which indicates that there is not much

decrease in makespan values at SchedSize = 73 at ClusterSize 23 and 24. So optimal makespan values for this SchedSize value lies near to edge when ClusterSize=23 for all scheduling policies. This is also an indicator that even if cluster size is further increased at SchedSize =73, no significant improvement in terms of makespan time can be observed.
Makespan prediction model equations for the three scheduling algorithms in terms of actual values of the SchedSize and ClusterSize are given in equations (12)-(14).

Makespan time for scheduling policy FCFS = 29.5907 – 0.1293 SchedSize + 0.1925 ClusterSize – 0.0101 (SchedSize x ClusterSize) + 0.0025 $SchedSize^2$ (12)

Makespan time for scheduling policy FPFS = 27.0598 – 0.1293 SchedSize + 0.1925 ClusterSize – 0.0101 (SchedSize x ClusterSize) + 0.0025 $SchedSize^2$ (13)

Makespan time for scheduling policy LJF = 32.0223 – 0.1293 SchedSize + 0.1925 ClusterSize – 0.0101 (SchedSize x ClusterSize) + 0.0025 $SchedSize^2$ (14)

These equations help to interpolate over the design space i.e. to predict the makespan time at new design points of input variables and these predictions need to be verified by few additional experimental runs as is done in table. This table shows the closeness of newly predicted values by RSM based second-order polynomial model equations and the experimentally computed makespan values.

Table 4: Validation results

Exp. No.	SchedSize	ClusterSize	SchedPolicy	Makespan (in seconds)			
				Exp.	RSM	MLP	RBF
37	73	18	FCFS	23.522	23.668	23.116	23.829
38	177	18	FCFS	55.975	56.314	55.015	57.796
39	73	23	FCFS	20.914	20.944	21.421	20.586
40	107	22	FPFS	22.665	22.307	21.722	48.397
41	177	22	FPFS	46.834	47.702	46.015	42.146
42	146	24	FPFS	30.176	30.702	29.412	30.543
43	73	20	LJF	23.976	25.010	23.774	26.919
44	177	20	LJF	55.675	55.555	55.164	51.462
45	177	22	LJF	51.941	52.364	51.839	51.462

3. ANN THEORY AND MODELING PROCEDURE

Artificial neural networks are the parallel computing tools that tend to simulate the behavior of neurons of human-brain. They are termed as universal approximators because of their learning capability to mimic any given functional relationship that can be computed by ordinary digital computer. ANNs are also known as the natural extensions and generalizations of statistical regression techniques due to their ability to model complex non-linear relationship existing between input and output response variables. Two forms of the feed-forward neural networks namely multi-layer perceptron (MLP) and radial basis functions (RBF) generally consist of three layer network architecture with each layer consists of number of neurons; input layer consisting of neurons representing input variables, output layer neurons indicating outcome and a hidden layer neurons comprise of hidden units in case of MLP and non-linear radial units in the case of RBF. The prime difference between MLP and RBF is in the number of hidden layers in the network architecture. RBF can only contain one hidden layer in the network architecture.

3.1 MLP model development

Input as well as output layer in the MLP network are of linear form and hidden layer consists of number of neurons representing non-linear activation function which is generally a sigmodial function. Each neuron carries out weighted sum of their inputs and passes this aggregated input through some activation function to produce output. An additional bias value may sometimes be added to the aggregated input to adjust the net input to be presented to the activation function. Neurons in the various layers are fully interconnected with each other except the neurons on the same layer. Output of a neuron in one layer can become input to the connected neuron in the other layer. The number of hidden neurons

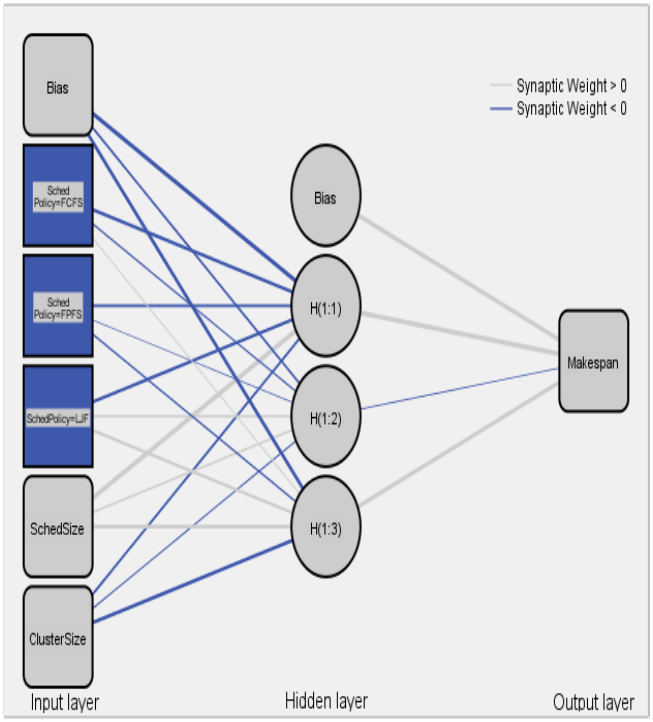

Figure 9: MLP neural network model

in the hidden layer is determined by iteratively varying the hidden neurons from 2 to 4 and the best results are observed when the hidden neurons are equal to 3. Hyperbolic tangent function is being used by the hidden neurons as the activation function to process weighted sum input to be passed to output layer neuron and identity function is used as the activation function by the output layer neuron to estimate output. The MLP network structure is shown in figure 9. MLP model is developed with the help of statistical data analysis software SPSS 17.0. Experimental data obtained from RSM phase with two inputs; SchedSize and ClusterSize are normalized between -1 and +1. Third input SchedPolicy is a categorical variable with three levels; it is treated as a fixed factor in SPSS 17.0 software. Experimental data is partitioned into three sets of distinct data values; training (53.3 %), test (26.7 %) and validation (20.0 %) data sets for training, testing

and validation of the MLP respectively. MLP networks are trained with training data (inputs and target output pairs) using scaled conjugate gradient (scg) optimization based back-propagation technique that is composed of two passes; forward pass and backward pass. In the forward pass, random values (between -1 and +1) are assigned to synaptic weights of various connection edges. MLP output i.e. makespan values are computed by the participating neurons based on the inputs, weights and activation functions. Difference between MLP computed output and a target output value is computed which is known as error value. In order to reduce this error to minimum possible extent a second pass of MLP training is required. In this backward pass, MLP learns by adjusting its synaptic weights after processing training data with number of epochs in order to minimize the squared error value averaged over all the training inputs. A single epoch (which is a cycle of representation of all training inputs patterns) comprises of applying all the input training patterns once and modifying weights at each step to minimize MSE value. Throughout each epoch, the error values are computed and propagated backwards to MLP network to adjust its synaptic weights. After each epoch, MSE is computed and if it exceeds the some small already specified value, then new epoch is started. MLP network might take number of epochs before it has learned the problem within acceptable MSE value. The reason for choosing 'scg' training algorithm is its ability to train large

Table 5: MLP network synaptic weights

Input layer to hidden layer weights		Hidden Layer 1		
		H(1:1)	H(1:2)	H(1:3)
Input Layer	(Bias)	-1.245	-0.318	-0.855
	(SchedPolicy=FCFS)	-0.385	-0.168	0.087
	(SchedPolicy=FPFS)	-0.330	-0.016	-0.200
	(SchedPolicy=LJF)	-0.378	0.269	0.345
	SchedSize	1.828	0.218	0.805
	ClusterSize	-0.327	-0.119	-0.433
Hidden layer to output layer weights		Output layer		
Hidden Layer 1	(Bias)	1.427		
	H(1:1)	1.592		
	H(1:2)	-0.075		
	H(1:3)	1.075		

networks, low memory requirements and faster convergence. The synaptic weights of various connection links between neurons of MLP network are given in table 5 and makespan prediction results are shown in table 2.

3.2 RBF neural network modeling

Radial basis function networks might require more neurons in the hidden layer than MLP networks due to the presence of only single hidden layer in the network architecture. Generally they can be trained much faster than MLP networks. The input as well as output layer in RBF networks are of linear type whereas the hidden layer contains non-linear radial units, each corresponding to only a local region of input space. The output layer executes a biased weighed sum of the radial units and performs an approximation of input-output mapping over the entire problem space.

The RBF procedure [14] trains the network in two steps:

1. The procedure determines the radial basis functions using clustering methods. The center and width of each radial basis

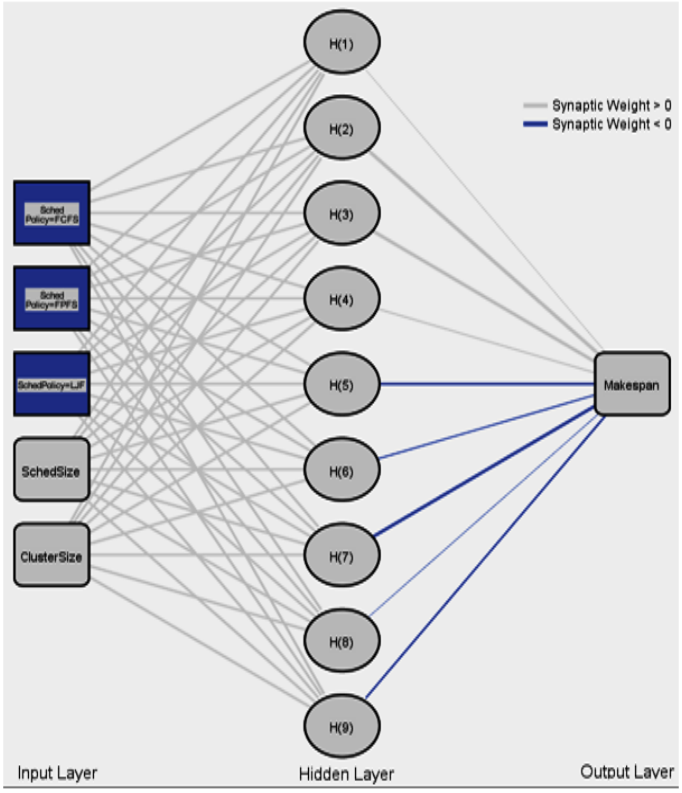

Figure 10: RBF neural network model

function are determined. Softmax is used as the normalized Gaussian radial basis function by hidden units.

2. The procedure estimates the synaptic weights given in the radial basis functions. The sum-of-squares error (MSE) function with identity activation function for the output layer is used for both prediction and classification. Ordinary least squares regression analysis is used to minimize the sum-of-squares error.

The structure of RBF network is given in figure 10. The number of hidden units is determined by the testing data criterion. The "best" number of hidden units is the one that yields the smallest error in the testing data. The makespan predictions done by RBF network model are shown in table 2.

3.3 Statistical performance measures

The various performance statistics used to compare the performance of three modeling techniques are given in table. Coefficient of determination i.e. R^2 is a fraction between 0 and 1 indicating the overall variation in the data accounted by the model. Values closer to 1 indicate the strong relationship between response variable and the combined linear predictor variable. Particularly for RSM model another standard statistical performance measure namely predicted-R^2 is considered as the value of R^2 might increase when even when some non-significant

term is added to the model. Predicted-R^2 estimates the amount of variation in the new data explained by the model. So predicted-R^2 is considered as better performance measure than R^2 in case of RSM model. Root mean square error (RMSE) is the square root of the residual mean square error. It measures the standard deviation associated with the experimental error and lack of fit. Another direct method used for describing the deviations is mean or absolute average deviation (AAD). AAD and MSE are calculated as below:

$$MSE = \frac{1}{N}\sum_{k=1}^{N}(y(x_k) - \hat{y}(x_k))^2$$

$$AAD = \left\{\frac{\sum_{k=1}^{N}(|y(x_k) - \hat{y}(x_k)|)}{n}\right\} X\ 100$$

where $\boldsymbol{y(x_k)}$ and $\hat{y}(\boldsymbol{x_k})$ are the experimental and model predicted values of makespan time respectively.

Table 6: Statistical model performance metrics

Model	R^2	RMSE	AAD
RSM	0.9968*	0.725	1.80%
MLP	0.9681	0.761	1.98%
RBF	0.9052	2.014	4.62%

Smaller values of RMSE and AAD are desired for all the predictive models. Results in the table show that the RSM tops among the three predictive models with high value of predicted- R^2 and small values of RMSE and AAD followed by MLP and RBF model.

4. CONCLUSION

This framework presents a set of empirical-methods to model the performance of space-sharing job scheduling algorithms in campus-wide PC-cluster environment. The RSM model not only helps to understand the relationship between significant input variables and the makespan using mathematical regression equations but also to tune input scheduling process parameters to achieve minimized makespan time. These regression models rank the input variables in terms of their significance in affecting the output. Feed-forward ANN forms viz. MLP and RBF models are trained with empirical data obtained with the help of experimental-design phase of RSM approach. After suitable training they can approximate the hidden functional relationship between input and output values of job scheduling algorithms. Generalization capability of these ANN models turn them an effective tool to predict or interpolate the performance of very large and complex distributed systems. Standard statistical error validation measures indicate that the makespan time predictions by the RSM model more closely approximates the experimental makespan values followed by MLP and RBF model.

5. REFERENCES

[1] Ismail, I.M.1995, "Space-sharing job scheduling policies for parallel computers", Ph.D thesis, Iowa state university, Iowa, 1995.

[2] Mohamed, A., Lester Lipsky and L., Ammar, R. 2003, "Performance Modeling of a Cluster of Workstations". In Proceedings of Communications in Computing'2003. pp.227~233.

[3] Figueira, S.M. 2004, "Optimal partitioning of nodes to space-sharing parallel tasks", *Parallel Computing*, 32(2004), 313-324.

[4] Iqbal,S., Gupta,R. and Fang,Y.C. 2005 "Planning considerations for job scheduling in HPC clusters", reprinted from Dell Power Solutions, pp 133-136.

[5] Schweigelshohn, U. and Yahyapour, R. 1998 "Analysis of first-come-first-serve parallel job scheduling", *In Proceedings of the ninth annual ACMSIAM symposium on discrete algorithms*, pages 629–638, Philadelphia, PA, Society for Industrial and Applied Mathematics.

[6] Aida, k. 2000, "Effect of job size characteristics on job scheduling performance" In Job Scheduling Strategies for Parallel Processing, Springer Verlag, Lect. Notes Computer Science vol. 1911, pp. 1—17.

[7] Sherwani, J., Ali, N., Lotia, N., Hayat, Z. and Buyya, R. "Libra: A Computational Economy based Job Scheduling System for Clusters", *Software: Practice and Experience*, vol. 34, no. 6, May 2004, pp.573-590.

[8] Benitez, N. and McSpadden, A. 1997, "Stochastic Petri Nets Applied to the Performance Evaluation of Static Task allocations in Heterogeneous Computing Environments," Proceedings of the 6th Heterogeneous Computing Workshop, pp. 185-194, 1997.

[9] K. Aida, H. Kasahara, and S. Narita.1998 "Job Scheduling Scheme for Pure Space Sharing Among Rigid Jobs", *In fourth workshop on Job Scheduling Strategies for Parallel Processing, Lecture Notes in Computer Science*, vol. 1459, pages 98-121, Springer-verlag.

[10] Goh, L.K. and Veeravalli, B.2008, "Design and performance evaluation of combined first-fit task allocation and migration studies in mesh multiprocessor systems", *Parallel Computing*, 34(9), 508-520, 2008.

[11] Collins, D. and George, A.2001, "Parallel and Sequential Job Scheduling in Heterogeneous Clusters: A Simulation Study Using Software in the Loop" in SIMULATION, november 2001, vol.77 no. 5-6,169-184.

[12] Rajaei, H., Dadfar, M. and Joshi, P.2006, "Simulation of job scheduling for small scale clusters", Proceedings of the 2006 Winter Simulation Conference.

[13] Montgomery, D.C. 2009 Design and analysis of experiments (5th ed.). New York: Wiley & Sons.

[14] SPSS 17.0 User's guide http://www.hks.harvard.edu accessed on October 2011.

[15] Anderson, M.J. and Whitcomb, P.J. 2005, RSM Simplified: Optimizing processes using response surface methods for design of experiments, CRC press.

[16] Antony, J. 2003. Design of experiments for engineers and scientists. Elseveir Science & Technology Books 2003.

[17] Myers, R.H., Montgomery, D.C. and Anderson-Cook, C. M., 2009. Response Surface Methodology: Process and product optimization using designed experiments (3rd ed.).New York: John Wiley and Sons, Inc. 728 pp.

[18] Anderson, M.J. and Whitcomb, P.J. 2000, *DOE Simplified: Practical Tools for Effective Experimentation*, Productivity press.

[19] Design Expert Software version 8.0 user's guide 2009.

[20] Haykins, S. Neural networks-A comprehensive foundation, Prentice Hall, 1999.