IJCTA, 9(22), 2016, pp. 347-357
© International Science Press

Statistical Model for Automated Database Tuning Framework

Hitesh Kumar Sharma* Abhinav Bhushan* Vaibhav Jain* Tanupreet Singh* Komal Munjal*

Abstract : Tuning a database is one of the main activities that help an application to run more quickly. The performance problem may be identified by slow or unresponsive system. This kind of problem usually occurs because high system load, causing some part of the system to reach a limit in its ability to respond. The limit of this system is called bottleneck. Today application using database are becoming complex as the size of data increasing. In the same way the performance related issues are also increasing. As DBMS are providing their configuration parameters to change their internal configuration but to change them in effective manner is tedious job and prone to error. To make it more effective and changing the configuration parameters for enhancing performance there is a strong need of automation. Managing 200+ parameter manually is not possible or it is a time consuming process. This paper explains the automated tuning framework and the algorithm to implement the proposed framework.

Keyword : Database Tuning, RDBMS, SGA Parameters, Correlation Coefficient, Coefficient of Variance.

1. INTRODUCTION

Tuning a database is one of the main activities that help an application to run more quickly. The performance problem may be identified by slow or unresponsive system. This kind of problem usually occurs because high system load, causing some part of the system to reach a limit in its ability to respond. The limit of this system is called bottleneck. Today application using database are becoming complex as the size of data increasing. In the same way the performance related issues are also increasing. As DBMS are providing their configuration parameters to change their internal configuration but to change them in effective manner is tedious job and prone to error. To make it more effective and changing the configuration parameters for enhancing performance there is a strong need of automation. Managing 200+ parameter manually is not possible or it is a time consuming process.

In this paper we have defined a solution to solve this complex problem (i.e. selection of appropriate resource to tune). We have formulated a mathematical solution for that and designed an algorithm to make this process automated.

2. AUTOMATED DATABASE TUNING FRAMEWORK

Today we are using the complex databases. Business requirement are also increased in the same ration. High performance and quick response is one of the major needs for a DBMS. Database can be tuned through physical design but continuously change in design is not possible for running application. As the physical design of database suffers from various limitations, an automated database tuning framework is proposed in order to achieve high grade of performance. This automated framework is used to identify the low performance system and alter the key parameters to enhance the same. The Framework has three basic building blocks:

* Centre for Information Technology, University of Petroleum & Energy Studies Dehradun, Uttarakhand, India hksharma@ddn.upes.ac.in

(*a*) Automated Workload Generation Block

(*b*) Working Database

(*c*) Automated Database Tuning Block

This framework is the combination of the above defined three blocks. After implementing this framework there will be no need for manual tuning. The combination of .net applications and database statistics an organization can achieve the automation in Database tuning. The complete collection of the above three components has been shown in following figures (Figure 1 & Figure 2).

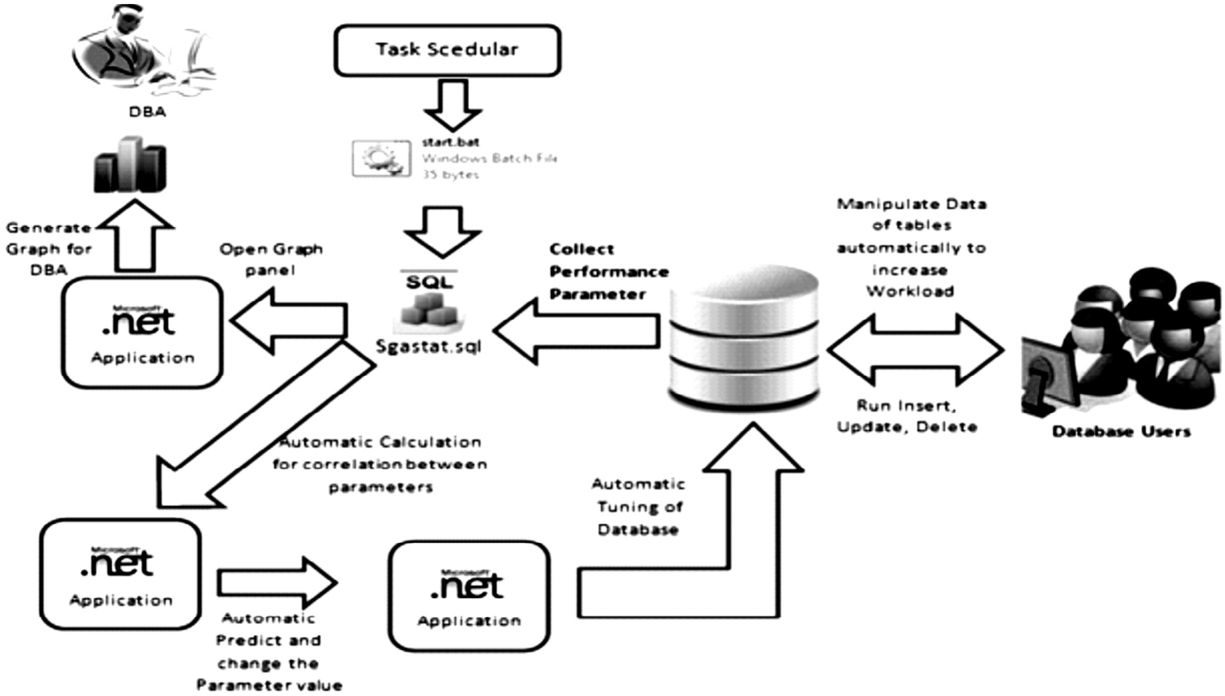
DBA
Task Scedular
start.bat
Windows Batch File
35 bytes
Generate Graph for DBA
Open Graph panel
Collect Performance Parameter
Manipulate Data of tables automatically to increase Workload
SQL
Sgastat.sql
.net
Application
Run Insert, Update, Delete
Database Users
Automatic Calculation for correlation between parameters
Automatic Tuning of Database
.net
Application
Automatic Predict and change the Parameter value
.net
Application

Fig. 1. Automated Database Tuning Framework.

The first figure uses the actual user workload and took the corrective measure. But in second figure the framework uses the virtual workload generated by the application itself by other .net component.

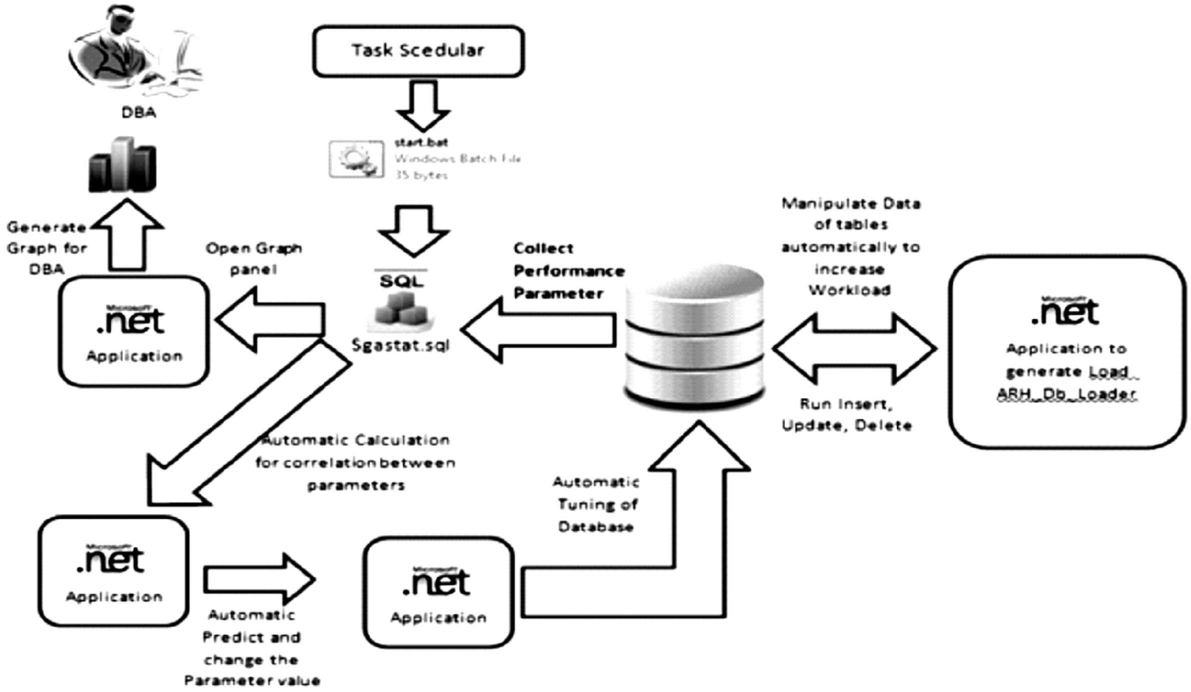
DBA
Task Scedular
start.bat
Windows Batch File
35 bytes
Generate Graph for DBA
Open Graph panel
Collect Performance Parameter
Manipulate Data of tables automatically to increase Workload
SQL
Sgastat.sql
.net
Application
.net
Application to generate Load ARH_Db_Loader
Run Insert, Update, Delete
Automatic Calculation for correlation between parameters
Automatic Tuning of Database
.net
Application
Automatic Predict and change the Parameter value
.net
Application

Fig. 2. Automated Database Tuning Framework with virtual workload.

If the DBA want to take a proactive measure on performance he/she can user the virtual workload generation module to predict the future performance related issue. Here we have integrated a .net application to make several connections to the database virtually.

3. MATHEMATICAL FORMULATION OF AUTOMATIC RESOURCE SELECTION PROCESS

This section describes the mathematical implementation to select the resources automatically which will make direct impact on performance. We present a new analysis method that effectively selects resources for automatic tuning in order to reduce the administrator's time, efforts, and intervention. The following subsections will define the two statistical coefficients (*i.e.* correlation coefficient and coefficient of variance) used in this paper.

A. Coefficient of Correlation

The word Correlation is made of Co- meaning "Together" and Relation meaning Dependency". Hence correlation coefficient explains the dependency of one variable on another variable.

Formula for Correlation coefficient :

$$\text{Corr\_Coff(X, Y)} = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \overline{X})^2 \sum_{i=1}^{n}(Y_i - \overline{Y})^2}} \qquad (1)$$

The relationship of one variable with other variable is decided on the basis of the following values of correlation coefficient

- + 1 (highly positive relationship)
- 0 (no relationship)
- – 1 (highly negative relationship)

The following graphical representation shows the relationship of two variables according to the values of correlation coefficient.

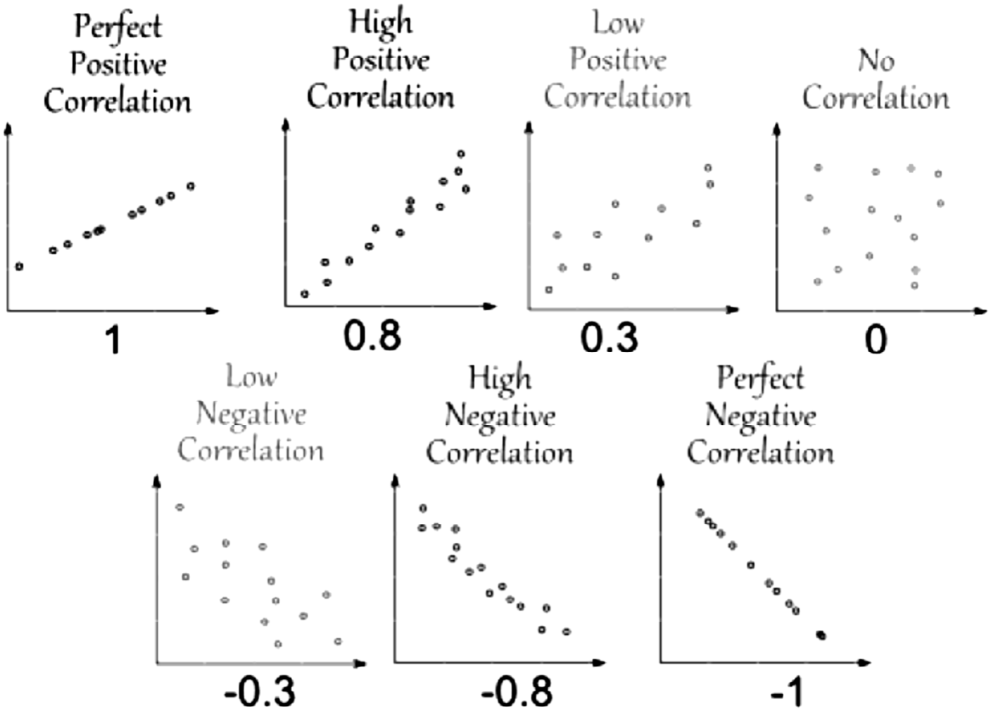

Fig. 1.

The perfect blue color shows the perfect positive relationship and the perfect red color graph shows the perfect negative relationship.

B. Variation Coefficient

The variation coefficient is used to calculate the variance of a data set. Based upon the value of this coefficient we can consider that how stable or variable the dataset is. To calculate this coefficient we need to calculate the mean and standard deviation for the given dataset.

Formula for Variance Coefficient :

$$\text{Coefficient of Variation (Y)} = \frac{\text{Standard Deviation } (\sigma)}{\text{Mean}(\overline{\text{Y}})} \tag{2}$$

4. USE OF CORRELATION COEFFICIENT AND COEFFICIENT VARIANCE FOR AUTOMATIC RESOURCE SELECTION

The above equations are used to find the impact of changing resource values on performance indicator. It may be positive, negative or no impact. To calculate the coefficient we need to find the mean and standard deviation of a list of observations for both. We should have equal number of observations for both resources as well as parameters.

In his paper we have coefficient of correlation is used to find the relation between resource value and parameters. The user needs to set a threshold value |t|. It will range from +1 to -1, if the value of this coefficient is +t or more than these will have positive relation. If this coefficient has _t value or less than there will be negative relation. Other values of this coefficient will provide no relationship.

Scenario 1. we have two lists of values for performance indicators P1 and P2. Suppose the threshold be |0.6| and a resource R1 (in megabytes) and the performance indicators P1 and P2 change as follows:

R1	64	128	192	256	320	384	448	512	576	640
P1	54.42	54.98	54.9	55.26	55.04	54.74	53.9	54.46	54.22	4.06
P2	148.64	153.58	156.5	161.26	163.38	1 63.9	168.6	169.22	175.4	178.82

Table 1 shows the middle steps used to calculate correlation coefficient between R1 and P1. The calculated value of correlation coefficient is given below to the corresponding table.

Table 1. Calculation of Correlation coefficient between R1 and P1.

R1	*P1*	*A = Ki-Mean (R1)*	*B = Pi-Mean(P1)*	*A*B*	*A^2*	*B^2*	*SQRT(M*N)*
64	54.42	-288	-0.178	51.264	82944	0.031684	
128	54.98	-224	0.382	-85.568	50176	0.145924	
192	54.9	-160	0.302	-48.32	25600	0.091204	
256	55.26	-96	0.662	-63.552	9216	0.438244	
320	55.04	-32	0.442	-14.144	1024	0.195364	793.046775
384	54.74	32	0.142	4.544	1024	0.020164	
448	53.9	96	-0.698	-67.008	9216	0.487204	
512	54.46	160	-0.138	-22.08	25600	0.019044	
576	54.22	224	-0.378	-84.672	50176	0.142884	
640	54.06	288	-0.538	-154.944	82944	0.289444	
Mean (R1)	Mean(P1)			O = SUM(A*B)	M = SUM(A^2)	N = SUM(B^2)	
352	54.598			-484.48	337920	1.86116	

Corr_Coff (R1,P1) = SQRT(M*N)/SUM(A*B) = -484.48/793.05= **– 0.61091**

Table 2 shows the middle steps used to calculate correlation coefficient between R1 and P1. The calculated value of correlation coefficient is given below to the corresponding table.

Table 2. Calculation of Correlation coefficient between R1 and P2.

R1	*P2*	*A = Ki-Mean (R1$^{)}$*	*B = Pi-Mean(P2)*	*A*B*	*A^2*	*B^2*	*SQRT(M*N)*
64	148.64	-288	-15.29	4403.52	82944	233.7841	
128	153.58	-224	-10.35	2318.4	50176	107.1225	
192	156.5	-160	-7.43	1188.8	25600	55.2049	
256	161.26	-96	-2.67	256.32	9216	7.1289	
320	163.38	-32	-0.55	17.6	1024	0.3025	16509.6815
384	163.9	32	-0.03	-0.96	1024	0.0009	
448	168.6	96	4.67	448.32	9216	21.8089	
512	169.22	160	5.29	846.4	25600	27.9841	
576	175.4	224	11.47	2569.28	50176	131.5609	
640	178.82	288	14.89	4288.32	82944	221.7121	
Mean (R1)	Mean(P2)			O = SUM(A*B)	M = SUM(A^2)	N = SUM(B^2)	
352	163.93			16336	337920	806.6098	

Corr_Coff (R1,P2) = SQRT(M*N)/SUM(A*B) = 16336/16509.68 = **0.98948**

Coefficient of correlation between R1 and P1 is about -0.61091. Coefficient of correlation between R1 and P1 is within the threshold, so there is no relationship. On other side R1 and P2 is about 0.98948. As the coefficient of correlation between R1 and P2 is over the threshold. So it has positive relationship. Coefficient of correlation is used to show only relationship between the resource and parameters. But it will not consider the variance of values. To get the effect of range of change, use another coefficient of variance, which is shown in equation 2? It provides a normalized value by calculating the standard deviation in means, especially when the number of data or measurement ranges is different. The variable value of the coefficient is important for change it indicate whether the change is in effective manner or not. The DBA need to set the threshold value as |z|. if this coefficient value is +z than the impact is positive is it is near to z or less than z then it will have useless impact.

Scenario 2. Here we have taken the list of two parameters P3 and P4. Suppose the threshold value is +0.6 or -0.6 for the coefficient of correlation and is 0.05 for the coefficient of variation.

R1	64	128	192	256	320	384	448	512	576	640
P3	149.66	146.6	145.68	146.62	142.16	139.26	140.88	139.54	140.44	138.32
P4	94.98	109.18	128.5	165.78	189.72	199.74	199.76	199.76	199.76	199.76

Table 3 & table 4 shows the middle steps used to calculate correlation coefficient between R1 and P3 and coefficient of variation for P3. The calculated value of correlation coefficient is given below to the corresponding table and the value of coefficient of variation is give in the last column of table 4

Table 3. Calculation of Correlation coefficient between R1 and P3

R1	*P3*	*A = Ki-Mean (K)*	*B = Pi-Mean(G)*	*A*B*	*A^2*	*B^2*	*SQRT(M*N)*
64	149.66	-288	6.744	-1942.272	82944	45.481536	
128	146.6	-224	3.684	-825.216	50176	13.571856	
192	145.68	-160	2.764	-442.24	25600	7.639696	
256	146.62	-96	3.704	-355.584	9216	13.719616	
320	142.16	-32	-0.756	24.192	1024	0.571536	6807.69622
384	139.26	32	-3.656	-116.992	1024	13.366336	
448	140.88	96	-2.036	-195.456	9216	4.145296	
512	139.54	160	-3.376	-540.16	25600	11.397376	
576	140.44	224	-2.476	-554.624	50176	6.130576	
640	138.32	288	-4.596	-1323.648	82944	21.123216	
Mean (R1)	Mean(P3)			SUM(A*B)	M=SUM(A^2)	N=SUM(B^2)	
352	142.916			-6272	337920	137.14704	

Co_cf (R1,P3) = SQRT(M*N)/SUM(A*B) = -6272/6807.70 = **-0.921310206**

Table 4. Calculation of coefficient of variance for G

P3	*B = $P3_i$-Mean(P3)*	*B^2*	*σ*	*Coff_Var(P3)*
149.66	3.372	11.370384		
146.6	1.842	3.392964		
145.68	1.382	1.909924		
146.62	1.852	3.429904		
142.16 139.26	-0.378 -1.828	0.142884 3.341584	1.851668437	0.012956341
140.88	-1.018	1.036324		
139.54	-1.688	2.849344		
140.44	-1.238	1.532644		
138.32	-2.298	5.280804		
Mean(P3)		N=SUM(B^2)		
142.916		34.28676		

Table 5 & table 6 shows the middle steps used to calculate correlation coefficient between R1 and P4 and coefficient of variation for P4. The calculated value of correlation coefficient is given below to the corresponding table and the value of coefficient of variation is give in the last column of table 6.

Table 5. Calculation of Correlation coefficient between R1 and P4.

R1	*P4*	*A = Ki-Mean (R1)*	*B = Pi-Mean(P4)*	*A*B*	*A^2*	*B^2*	*SQRT(M*N)*
64	94.98	-288	-73.714	21229.632	82944	5433.7538	
128	109.18	-224	-59.514	13331.136	50176	3541.9162	
192	128.5	-160	-40.194	6431.04	25600	1615.55764	
256	165.78	-96	-2.914	279.744	9216	8.491396	
320	189.72	-32	21.026	-672.832	1024	442.092676	
384	199.74	32	31.046	993.472	1024	963.854116	73221.9642
448	199.76	96	31.066	2982.336	9216	965.096356	
512	199.76	160	31.066	4970.56	25600	965.096356	
576	199.76	224	31.066	6958.784	50176	965.096356	
640	199.76	288	31.066	8947.008	82944	965.096356	
Mean (R1)	Mean(P4)			SUM(A*B)	M=SUM(A^2)	N=SUM(B^2)	
352	168.694			65450.88	337920	15866.0512	

Co_cf(R1,P4) = SQRT(M*N)/SUM(A*B) = 65450.88/73221.96 = **0.893869493**

Table 6. Calculation of coefficient of variance of P4

P4	*B = Pi-Mean(P4)*	*B2*	*σ*	*Coff_Var(P4)*
94.98	-36.857	1358.438449		
109.18	-29.757	885.479049		
128.5	-20.097	403.889409		
165.78	-1.457	2.122849		
189.72	10.513	110.523169	19.91610607	0.118060548
199.74	15.523	240.963529		
199.76	15.533	241.274089		
199.76	15.533	241.274089		
199.76	15.533	241.274089		
199.76	15.533	241.274089		
Mean(H)		N = SUM(B2)		
168.694		3966.51281		

The Coefficient of Correlation between R1 and P3 is about -0.92131, and the coff. Of variation of P3 is about 0.012956341. we can say that there is no relation between P3 and R1 because the coefficient of variation is below 0.05, coefficient for correlation is below -0.6 but it will not be considered. The coefficient of correlation between R1 and P4 is about 0.893869, and the coefficient of variation of P4 is about 0.118060. Since the correlation coefficient between R1 and P4 is above +0.6 and coefficient of variation of P4 is over 0.05, we say that these have positive relation

5. ALGORITHM FOR AUTOMATIC SELECTION OF RESOURCE

As we have seen above, by using some statistics coefficients (*i.e.* Correlation Coefficient and Coefficient of Variance) we can find the positive, negative or no relation between resource parameters and performance indicators. But the manual calculation of these parameters is again a tedious task. To overcome this issue we have designed a set of algorithms. These algorithms can be implemented Varo a small computer application using any computer programming language. After implementation of these algorithms, the process of automatic selection of the resources responsible for good performance will be automated. The set of algorithms contain three algorithms the algorithms with their significance have been explained separately in coming subsections.

A. Algorithm 1: (Algorithm to calculate Correlation Coefficient)

This algorithm will calculate the correlation coefficient between resource parameter and performance indicator. The algorithm takes two arrays X[ ], Y[ ] as input. The array X[ ] will contain some values for a particular resource and the array Y[ ] will have some value of indicator corresponding to each value of resource.

```
Co_cf(X[ ],Y[ ])
{
Var s_x=0;
Var s_y=0;
Var mean_x;
Var m_y;
Var F_nm=0;
Var D_x=0;
Var D_y=0;
Var F_D;
Var Co_cf;
Var n= X.len
for(Var i=0; i<n;i++)
{
s_x = s_x + X[i];
s_y = s_y + Y[i];
}
mean_x=s_x/n;
m_y=s_y/n;
for(Var i=0; i<n;i++)
{
F_nm = F_nm + ((X[i]-m_x)*(Y[i]-m_y));
}
for(Var i=0; i<n;i++)
{
D_x = D_x + ((X[i]-m_x)*(X[i]-m_x));
D_y = D_y + ((Y[i]-m_y)*(Y[i]-m_y));
}
F_D= sqrt(D_x * D_y);
Rel_Coff= F_nm/ F_D;
Return Co_cf;
}
```

The output of the algorithm will be the value of correlation coefficient between X[ ](*i.e*. array of resource values) and Y[ ] (*i.e*. array of performance indicator values) .

B. Algorithm 2 (Algorithm to calculate Coefficient of Variation)

The following algorithm calculates the coefficient of variation for a given indicator. The parameter array is passed to this algorithm.

```
Var_Coff(Y[ ])
{
Var m_y;
Var nm=0;
Var s_y=0;
Var n= Y.lenght;
Var std_dev;
Var var_coff;
for(Var i=0; i<n;i++)
{
s_y = s_y + Y[i];
}
m_y=s_y/n;
for(Var i=0; i<n;i++)
{
nm = nm + ((Y[i]-m_y)*(Y[i]-m_y));
}
std_dev=sqrt(nm/n);
var_coff= std_dev/m_y;
Return var_coff;
}
```

The output of this algorithm will be the value of the coefficient of variance for the input array Y[ ] (i.e. the array of indicator values).

C. Algorithm 3 (Algorithm for resource selection)

It is the final algorithm for the automation. It takes four parameters as input and these parameters are given below.

- An array of resource vales and names name (*i.e. a_rs*[ ][ ])
- An array of Parameters vales and names name (*i.e. a_p*[ ][ ])
- Coefficient of correlation value (*i.e*. th_CC)
- Coefficient of variation value (*i.e*. th_CV)

These values will be passed to the following algorithm and this algorithm will automatically calls the above two algorithms by passing the required parameters. The output of above algorithms will be used as input for next steps of this algorithm.

```
Select_Tuning_Resource (a_rs[ ][ ],a_p[ ][ ],th_CC,th_CV)
{
Var n = ar_rs.nr;
Var m= ar_rs.nc;
Var a_rs_1D[ ];
Var a_p_1D[ ];
Var co_cf[n];
Var var_coff[n];
for(Var i=0;i< n;i++)
{
                    for(Var j=0;j< m;i++)
{
                   a_rs_1D[j]= a_rs[i][j];
a_p_1D[j]= a_p[i][j];
}
Co_cf[i]= Co_cf(a_rs_1D,a_p_1D);
        var_coff [i]= Var_Coff(a_p_1D);
}
for(Var i=0;i< n;i++)
{
If(Co_cf[i]> th_CC &&  var_coff [i]> th_CV)
{
print  "It has positive impacte";
}
Else If(Co_cf[i]< th_CC &&  var_coff [i]> th_CV)
{
print "It has negative impact";
}
                                      Else
{
print  "There is no impact";
}
}
}
```

6. SUMMERY & FUTURE WORK

The proper management of the resources is one of the major parts in database tuning. In this paper we have focused on the selection of the major resource responsible for high/low performance. Before changing the value of a parameter it highly recommended to find its positive or negative impact on performance. Majorly this task is being done by a DBA but in this paper we have proposed some algorithms that will automatically populate the list of the resources which will have positive impact on performance after manipulate them. On implementing these algorithms and convert them Varo an application anyone can change the right resources to get better performance. In future there is a need for implement this work using a programming language to Varegrate these algorithm in working RDBMS tools.

7. REFERENCES

1. Edward Whalem, Oracle performance tuning and optimization. SAMS publication.
2. Xu, X., Martin, P. and Powley, W., “Configuring Buffer Pools in DB2 UDB”, IBM Canada Ltd., the National Science and Engineering Research Council (NSERC) and Communication and Information Technology Ontario (CITO), 2002.
3. Chaudhuri, S. (ed). Special Issue on, “Self-tuning Databases and Application Tuning”, IEEE Data Engineering, *Bulletin 22(2)*, June 1999.
4. Bernstein, P. *et al.*, “The Asilomar Report on Database Research”, ACM SIGMOD Record 27(4), December 1998, pp. 74 - 80.
5. Nguyen, H. C., Ockene, A., Revell, R., and Skwish, W. J., “The role of detailed simulation in capacity planning”. IBM Syst. J. 19, 1 (1980), 81-101.
6. Seaman, P. H., “Modeling considerations for predicting performance of CICS/VS systems”, IBM Syst. J. 19, 1 (1980), 68-80.
7. Foster, D. V., Mcgehearty, P. F., Sauer, C. H., and Waggoner, C. N., “A language for analysis of queuing models”, *Proceedings of the 5th Annual Pittsburgh Modeling and Simulation Conference* (Univ. of Pittsburgh, Pittsburgh, Pa., Apr. 24-26). 1974, pp. 381-386.
8. Reiser, M., and Sauer, C. H., “Queuing network models: Methods of solution and their program implementation”, *Current Trends in Programming Methodology*. Vol. 3, Software Modeling and Its Impact on Performance, K. M. Chandy and R. T. Yeb, Eds. Prentice-Hall, Englewood Cliffs, N. J., 1978, pp. 115-167.