Effective Cost Models for Predicting Web Query Execution Cost

Shashidhara H R, Sanjay P K, G T Raju, Vinayaka Murthy

Abstract—Classical query optimizers rely on sophisticated cost models to estimate the cost of executing a query and its operators. By using this cost model, an efficient global plan is created by the optimizer which will be used to execute a given query. This cost modeling facility is difficult to be implemented in Web query engines because many local data sources might not be comfortable in sharing meta data information due to confidentiality issues.

In this work, an efficient and effective cost modeling techniques for Web query engines are proposed. These techniques does not force the local data sources to reveal their meta data but employs a learning mechanism to estimate the cost of executing a given local query. Two cost modeling algorithms namely: Poisson cost model and Exponential cost model algorithms are presented. Empirical results over real world datasets reveal the efficiency and effectiveness of the new cost models.

Keywords–Cost models, web query optimization, mediator, operators

I. INTRODUCTION

Web applications [1] have become a major tool in storing and accessing data. The data sources of Web applications are often located in different geographical locations. The Web application integrates these different data sources and builds an user required application. This framework even-though has provided an opportunity to integrate different data sources with minimal overhead, it suffers from performance bottlenecks due to bloated response times [2]. Consider a Web application which provides users information about properties in a particular city. Now the details of each real estate company and their corresponding property information is stored in their respective databases. The owners of such data sources might not be willing to migrate their entire data information due to confidentiality reasons. So, a Web application can be built by integrating these data sources without migrating the local data [2].

The Web application is built over a Web query engine shown in Figure 1 which is responsible for the execution of the query and providing the required answers. The Web application interface is connected with a component called as the Mediator. This Mediator manages the execution of the Web query. It divides the Web query into a group of local queries which will be mapped to the required local data sources. The local data sources also have a component called the Wrapper which acts as a interface between the local data source and the Mediator. The Mediator presents the local query to the Wrapper which executes this query. The executed result is converted to the desired format as specified by the Mediator. Then, Mediator combines the results from different Wrappers and joins the result to be presented to the user of the Web application.

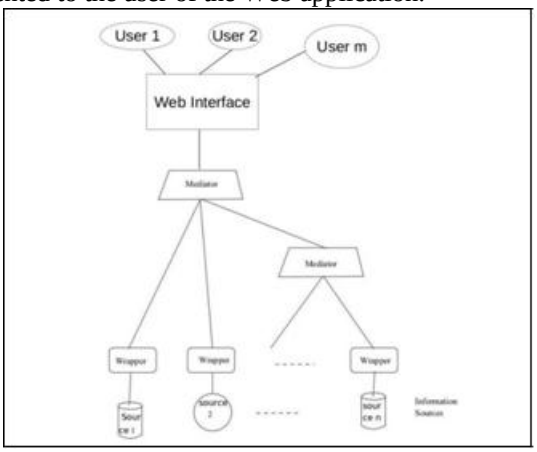

Fig. 1: Web Query Engine

The classical query optimizers use sophisticated cost models to compute the cost of executing a particular query. This query cost is composed of individual cost of the operators that are part of the final query plan. Each query might generate multiple plans and the minimum cost plan is selected for query execution. The same facility is absent in Web query engines and it has remained an open issue [3]–[6]. The reason being, local data sources have their own individual operator/plans to execute a given local query. Local data sources might not be comfortable in sharing this execution plan information with the Mediator. Due to the absence of cost information the Mediator is handicapped to produce an efficient global plan. So, the mediator might produce an inefficient plan which can result in bloated response time for the Web application user thereby, increasing the dissatisfaction of the user in adapting that Web application. So, it is crucial to develop some efficient and effective techniques to model the cost of executing a local query without relying on the local data sources to provide that information.

In this work, the problem of developing cost models for Web query execution engine is addressed and the following contributions are made:

1. Empirical investigations were conducted to determine the best fitting cost model. Two cost models were found suitable to be applied on this problem. They are, Poisson cost model and the Exponential cost model.
2. The Poisson cost model is illustrated by describing the parameter estimation technique and cost model calculation algorithm.

Revised Manuscript Received on December 12, 2019.

*Correspondence Author

Dr. Shashidhara H R*, Associate Professor, Department of Computer Science & Engineering, RNS Institute of Technology, Bengaluru. Email: shashi_dhara@yahoo.com

Sanjay P K, Assistant Professor, Department of Computer Science & Engineering, RNS Institute of Technology, Bengaluru. Email: sanin009@yahoo.com

G T Raju, Department of Computer Science & Engineering, RNS Institute of Technology, Bengaluru.

Vinayaka Murthy, Department of Computer Science & Engineering, RNS Institute of Technology, Bengaluru.

3. Similarly, the Exponential cost model is presented with its parameter estimation technique and cost model calculation algorithm.
4. A model selection framework is proposed to select one of these 2 techniques, which provides the best accuracy in execution cost estimation.
5. Empirical validation is performed on DBLP dataset. The effectiveness of these cost models are exhibited.

II. RELATED WORK

There are 4 frameworks for designing Web Query execution engines. The cost based framework [7]–[10] produces the best plan which has the minimum cost among the set of competing plans. This framework has a similar design when compared with classical database optimizers. The other 3 remaining frameworks deal with the result quality [11], [12], failure adaptability [13]–[15] and data source quality [16]–[19]. These 3 frameworks do not aim to produce the minimum cost plan but, provide other functionalities such as, better quality of result, recovery from system failures and analyzing the properties of local data sources.

Effective cost models for Web query execution engine are still elusive. Lack of cooperation from the local data sources has lead to the design of ineffective techniques which can suffer from frequent bloated response time problem [7]–[10]. Until, effective cost models are designed the Web query execution engine will continue to suffer from performance bottlenecks.

III. PROBLEM FRAMEWORK

Let, x and y be the number of tuples retrieved and the execution cost in seconds for a local query Q executed at the local data source L. The training set is given by, *Training Set* = $[x_1y_1, x_2y_2,...x_ny_n]$. Here, x_j and $y_j (1 \leq j \leq n)$ be the number of tuples retrieved and the execution cost in seconds for a local query Q_j. The task is to compute the execution cost $y\hat{}_i$ for a non training set query Q_i which has an estimated number of tuples $x\hat{}_i$.

IV. POISSON COST MODEL

The Poisson cost model is developed by using the Poisson distribution. For a random variable y, the Poisson density function is illustrated in Equation 1. The expected value for y is $E(y) = \mu$ and variance of y is $Var(y) = \mu$.

$$f(y) = \frac{e^{-\mu}\mu^{y}}{y!} \qquad y = 0, 1, 2.... \tag{1}$$

The regression function which models the relationship between x_i and y_i is shown in Equation 2.

$$y_i = E(y_i) + \varepsilon_i \qquad i = 1,2....n \tag{2}$$

Here, $E(y_i) = \mu_i$.

The link function $g()$ and its relation with μ_i is illustrated in Equations 3 and 4.

$$g(\mu_i) = \eta_i = \beta_0 + \beta_1 x_1 + .. + \beta_k x_k = \boldsymbol{x_i'\beta} \tag{3}$$

$$\mu_i = g^{-1}(\eta_i) = g^{-1}(\boldsymbol{x_i'\beta}) \tag{4}$$

If $g()$ is an identity link then, its relationship with μ_i is shown in Equation 5.

$$\mu_i = g(\mu_i) = \boldsymbol{x_i'\beta} \tag{5}$$

If $g()$ is a log link then, its relationship with μ_i is shown in Equation 6 and 7.

$$\log \mu_i = g(\mu_i) = \boldsymbol{x_i'\beta} \tag{6}$$

$$\begin{aligned} \mu_i &= g^{-1}(\boldsymbol{x_i'\beta}) \\ &= e^{\boldsymbol{x_i'\beta}} \end{aligned} \tag{7}$$

The parameter β needs to be estimated from the likelihood function shown in Equation 8.

$$\begin{aligned} L(\boldsymbol{y}, \beta) &= \prod_{i=1}^{n} f_i(y_i) \\ &= \prod_{i=1}^{n} \frac{e^{-\mu_i}\mu_i^{y_i}}{y_i!} \\ &= \frac{\prod_{i=1}^{n} \mu_i^{y_i} \exp\left(-\sum_{i=1}^{n} \mu_i\right)}{\prod_{i=1}^{n} y_i!} \end{aligned} \tag{8}$$

The log likelihood function shown in Equation 9 is maximized w.r.t the parameter β. The parameter β obtained by this maximization procedure will be its estimated value which will be denoted as $\hat{\beta}$.

$$\boxed{\log L(\boldsymbol{y}, \beta) = \sum_{i=1}^{n} y_i \log(\mu_i) - \sum_{i=1}^{n} \mu_i - \sum_{i=1}^{n} \log(y_i!)} \tag{9}$$

So, the regression function describing the relationship between $y\hat{}_i$ and $\hat{\beta}$ is described in Equation 10.

$$\hat{y}_i = g^{-1}(\boldsymbol{x_i'\hat{\beta}}) \tag{10}$$

Finally, the estimated value of y_i by using identity link function is given in Equation 11 and by using log link function is given in Equation 12.

$$\boxed{\hat{y}_i = \boldsymbol{x_i'\hat{\beta}}} \tag{11}$$

$$\boxed{\hat{y}_i = \exp(\boldsymbol{x_i'\hat{\beta}})} \tag{12}$$

If, instead of x_i its estimated value $x\hat{}_i$ is used then, the Equations 11 and 12 become as shown in Equations 13 and 14.

$$\hat{y_i} = \hat{x_i'}\hat{\beta} \quad (13)$$

$$\hat{y_i} = \exp(\hat{x_i'}\hat{\beta}) \quad (14)$$

The Algorithm 1 describes the procedure to estimate the cost of running a local query over a given local data source. In the pre processing module, the parameter $\hat{\beta}$ is estimated by log likelihood function maximization through the parameter β. The value of β that maximizes this log likelihood function will become the estimated value $\hat{\beta}$. During query execution stage, the mediator system needs to calculate the cost of executing the local query. So, it estimates the cost $\hat{y}_i$ by using Equation 13 or Equation 14. The mediator system uses this cost information to build an efficient global plan for executing the Web query. After the executing, the actual number of tuples in result set of L_i and its execution cost is updated in training set to recalculate the parameter $\hat{\beta}$ for future cost calculation.

Algorithm 1 Poisson Cost Model Algorithm

[Pre-Processing Step]

Calculate the parameter $\hat{\beta}$ from the training set by maximizing the log likelihood function given in Equation 9 w.r.t parameter β.

[Query Execution Module]

Let L_i be a local query provided to a local data source by the Mediator system.

Let $\hat{x}_i$ be the number of tuples that can be retrieved for L_i. Calculate the estimated cost of executing L_i by using the Equation 13 or Equation 14.

Send the estimated cost $\hat{y}_i$ to the mediator system.

[Post-Processing Module]

After executing L_i, update the information about the actual cost of execution y_i and actual result set size x_i to the training set. Re perform the Pre-Processing Module.

V. EXPONENTIAL COST MODEL

The exponential cost model is built over the double exponential distribution shown in Equation 16. The regression model shown in Equation 15 can also be termed as robust regression model because unlike classical regression model, it does not assume normal observations of the training set.

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i = \boldsymbol{x_i'\beta}, \quad i = 1, 2, ..n \quad (15)$$

$$f(\varepsilon_i) = \frac{1}{2\sigma} e^{-\frac{|\varepsilon_i|}{\sigma}}, \quad -\infty < \varepsilon_i < \infty \quad (16)$$

The likelihood function to estimate the parameters β_0 and β_1 is shown in Equation 17. This involves minimizing the errors $\sum_{i=1}^{i=n} \varepsilon_i$. But, the regression model does not assume normal errors and hence, least square parameter estimator used in maximizing the likelihood function fails to provide good estimates because it requires normal error distribution. So, robust estimators shown in Equation 18.

$$L(\beta_0, \beta_1) = \prod_{i=1}^{n} \frac{1}{2\sigma} e^{-\frac{|\varepsilon_i|}{\sigma}} = \frac{1}{(2\sigma)^n} \exp\left(-\sum_{i=1}^{n} \frac{|\varepsilon_i|}{\sigma}\right) \quad (17)$$

$$\min_{\beta} \sum_{i=1}^{n} \rho(e_i) = \min_{\beta} \sum_{i=1}^{n} \rho(y_i - \boldsymbol{x_i'\beta}) \quad (18)$$

Equation 19 is an scale invariant version of Equation 18.

$$\min_{\beta} \sum_{i=1}^{n} \rho\left(\frac{e_i}{s}\right) = \min_{\beta} \sum_{i=1}^{n} \rho\left(\frac{y_i - \boldsymbol{x_i'\beta}}{s}\right) \quad (19)$$

The parameter s is estimated according to Equation 20.

$$s = \frac{median|e_i - median(e_i)|}{0.6745} \quad (20)$$

To minimize Equation 19, the first partial derivatives of ρ w.r.t $\beta_j (j = 0,1,..k)$ are equated with 0 which provides the required conditions for minimization. This results in a system of $p = k + 1$ equations shown in Equation 21.

$$\sum_{i=1}^{n} x_{ij} \psi\left(\frac{y_i - \boldsymbol{x_i'\beta}}{s}\right) = 0 \quad (21)$$

The Equation 21 can be rewritten as Equation 22 where the weights w_{i0} are given by Equation 23.

$$\sum_{i=1}^{n} x_{ij} \psi\left(\frac{y_i - \boldsymbol{x_i'\beta}}{s}\right) = \sum_{i=1}^{n} x_{ij} w_{i0} (y_i - \boldsymbol{x_i'\beta}) = 0, \quad j = 0, 1, ..k \quad (22)$$

$$w_{i0} = \begin{cases} \frac{\psi[(y_i - \boldsymbol{x_i'\hat{\beta}_0})/s]}{(y_i - \boldsymbol{x_i'\hat{\beta}_0})/s} & if\ y_i \neq \boldsymbol{x_i'\hat{\beta}_0} \\ 1 & if\ y_i = \boldsymbol{x_i'\hat{\beta}_0} \end{cases} \quad (23)$$

By using matrix notation Equation 22 becomes as shown in Equation 24.

$$\boldsymbol{X'W_0X\beta = X'W_0y} \quad (24)$$

The estimated parameter $\hat{\beta}$ is shown in Equation 25.

$$\hat{\beta} = (\boldsymbol{X'W_0X})^{-1}\boldsymbol{X'W_0y} \quad (25)$$

The Algorithm 2 describes the procedure for calculating the cost of executing a local query using the exponential cost model. This algorithm works on the similar lines of Algorithm 1 but instead of maximizing the likelihood function, minimization of robust regression function shown in Equation 19 is performed w.r.t β to obtain the estimated parameter $\hat{\beta}$. Finally, the cost of executing the local query $\hat{y}_i$ is estimated by using the parameter β.

Algorithm 2 Exponential Cost Model Algorithm

[Pre-Processing Step]

Calculate the parameter $\hat{\beta}$ from the training set by minimizing the robust regression function given in Equation 19 w.r.t parameter β.

[Query Execution Module]
Let L_i be a local query provided to a local data source by the Mediator system.
Let $\hat{x}_i$ be the number of tuples that can be retrieved for L_i. Calculate the estimated cost of executing L_i by using the Equation 15 in which x_i is replaced by its estimated value $\hat{x}_i$ and parameter $\hat{\beta}$ is calculated by Equation 25.
Send the estimated cost $\hat{y}_i$ to the mediator system.
[Post-Processing Module]
After executing L_i, update the information about the actual cost of execution y_i and actual result set size x_i to the training set. Re perform the Pre-Processing Module.

V. MODEL SELECTION FOR LOCAL QUERY COST ESTIMATION

Since, 2 models have been proposed in this work for local query cost estimation, it is important to guide the Mediator system to select the suitable model, depending upon the distribution of data, and data storage format inside the local data sources. To quantify the most suitable model to predict cost of executing a local query Q, Equation 26 is utilized. The metric $model\_score(m_i)$ provides the score of model m_i to predict execution cost of Q. Lower the score, better will be the accuracy for the model m_i to predict execution cost of Q. Here, the predicted execution cost by model m_i is indicated by x_j, and cost x_j represents actual execution for r previously executed queries, which are utilized to calculate $model\ score(m_i)$.

$$model\_score(m_i) = \frac{\sum_{j=1}^{r} |\overline{x_j} - x_j|^2}{r} \quad (26)$$

Algorithm 3 illustrates the model selection algorithm for estimating the execution cost of local query. The model scores for both models namely: Poisson Cost estimator and Exponential Cost estimator are calculated by utilizing predicted cost and actual cost for previously executed queries. The model which has the lowest score is utilized for calculating the estimated execution cost of next query Q This process is repeated for every new local query that needs to be executed.

Algorithm 3 Model Selection Algorithm for Local Query Cost Estimation

---

Let, P_c and E_c represent Poisson Cost estimator and Exponential Cost estimator respectively.
Let, r represent the number of previously executed queries, where x_j represents the actual execution cost of the j^{th} previously executed query. Let, $est(Q_j,P_c)$ and $est(Q_j,E_c)$ represent the estimated cost of j^{th} query by P_c and E_c respectively.
Calculate $model\_score(P_c) = \frac{\sum_{j=1}^{r} |est(Q_j,P_c) - x_j|^2}{r}$
Calculate $model\_score(E_c) = \frac{\sum_{j=1} |est(Q_j,E_c) - x_j|}{r}$
Select the model having the lowest model score, and utilize it for calculating the estimated execution cost for Q.
Also calculate the estimated execution cost for Q by using the model which was not selected. This statistic will be later used for future model selection procedure.

VI. EXPERIMENTS

The DBLP dataset is used for empirical study to demonstrate the performance efficiency of the proposed techniques for cost modeling. The dataset has a size of 650 *mb*. The Web Query Engine was simulated by dividing each table in the DBLP dataset into simulated local data sources. Some of the tables were given exclusive security and autonomous privileges. Also, a modified DBLP dataset was created by injecting skew into the original tables. This skew version helps in evaluating the robustness of the new cost modeling techniques. The performance study involved both Poisson cost model algorithm(PCM) and Exponential cost model algorithm(ECM).

The first empirical study evaluates the performance of the 2 cost modeling techniques against the actual runtime costs. In Figure 2, both PCM and ECM perform similarly. This is because both the models have a good fitting for the cost estimation problem.

The cost modeling techniques have a tendency to perform poorly in the presence of skew. So, the next empirical analysis shown in Figure 3 evaluates the robustness of PCM and ECM techniques. As seen in Figure 3, both techniques demonstrate considerable performance robustness in the presence of skew.

The analysis of cost of executing a query by varying the query result size is shown in Figures 4 and 5. The query result size has little influence on the estimated cost quality of the new cost model techniques.

The influence of database size on predicted costs of the new cost models are analyzed in Figures 6 and 7. This analysis involves, executing 2 queries on different sizes of the same database. As seen in Figures 6 and 7, the new cost models exhibit performance effectiveness even when there is variation in the number of tuples inside the underlying database.

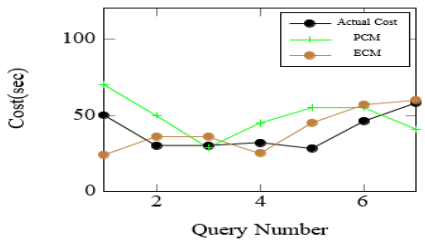

Fig. 2: Cost vs Query (DBLP)

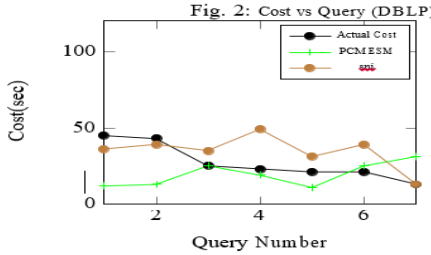

Fig. 3: Cost vs Query (Skew DBLP)

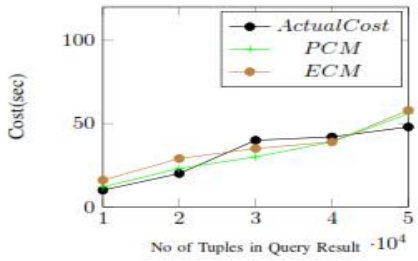

Fig. 4: Cost vs No Of Tuples($DBLP$)

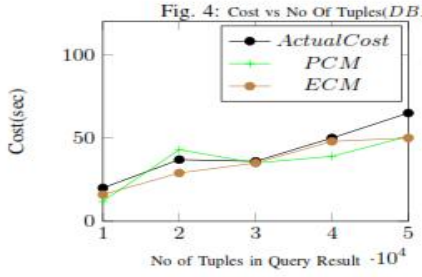

Fig. 5: Cost vs No of Tuples ($Skew\ DBLP$)

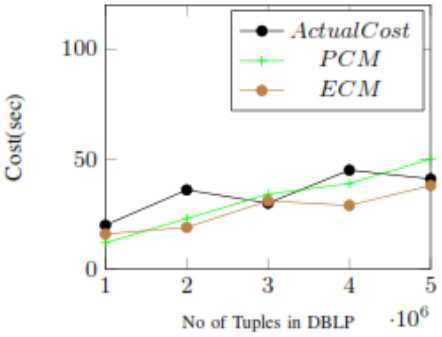

Fig. 6: DB Size vs Cost

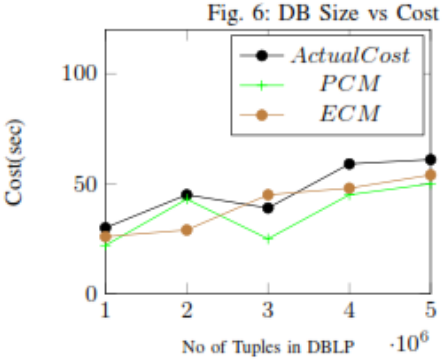

Fig. 7: DB Size vs Cost

VII. CONCLUSION

In this work, the problem of estimating the execution cost of a local query that needs to be executed at a local data source is addressed. Two techniques namely: Poisson cost modeling technique and exponential cost modeling technique is proposed. Both the approaches have proved their effectiveness as seen in the empirical results. These 2 models have been selected after thorough empirical evaluations on many real world datasets such as IMDB, Wikipedia and DBLP. Currently, these cost models can be directly used in many real world Web applications. In future, if some new dataset exhibits that these models are not suitable then, new model selection and evaluation is required.

In future, better cost models can be built which will use the partial meta data information that can be procured by the local data sources. Also, integrating the classical cost modeling technique and the Web query engine cost modeling technique would prove beneficial.

REFERENCES

1. T. Berners-Lee, J. Hendler, and O. Lassila, *The semantic Web* Scientific American, vol. 284, no. 5, 2001.
2. E. Bertino and A. Bouguettaya, *Introduction to the special issue on database technology on the Web*, IEEE Internet Computing, vol. 6, no. 4, 2002.
3. A. Ruiz, R. Corchuelo, Duran, and M. Toro, *Automated support for quality requirements in web-based systems*, in Proceedings of the 8th IEEE Workshop on Future Trends of Distributed Computing Systems, Bologna, Italy, IEEE, Oct.–Nov. 2001.
4. G.O. Arocena and A.O. Mendelzon, *WebOQL: Restructuring documents, databases and Webs*, in Proceed-ings of the 14th International Conference on Data Engineering, Orlando, Florida.
5. C. Batini, M. Lenzerini, and S.B. Navathe, *A comparative analysis of methodologies for database schema integration*, ACM Computing Surveys
6. T. Berners-Lee, *Services and Semantics: Web Architecture*, http://www.w3.org/2001/04/30-tbl, 2001.
7. A. Tomasic, L. Rashid, and P. Valduriez, *Scaling heterogeneous database and design of DISCO*, in Proceed- ings of the 16th International Conference on Distributing Computing Systems (ICDCS), Hong Kong, May 1996.
8. L.M. Haas, D. Kossmann, E.L. Wimmers, and J. Yang, *Optimizing queries across diverse data sources*, in Proceeedings of the 23rd International Conference on Very Large Data Bases (VLDB), Athens, Greece, Aug. 1997.
9. J.L. Ambite and C.A. Knoblock, *Flexible and scalable*
10. *query planning in distributed and heterogeneous environments*, in Proceedings of the Fourth International Conference on Artificial Intelligence Planning Systems, Pitsburg, USA, June 1998
11. S. Adali, K.S. Candan, Y. Papakonstantinou, and V.S. Subrahmanian, *Query caching and optimization in distributed mediator systems*, in Proceeedings of ACM SIGMOD International Conference on Management of Data, Montreal, Canada, June 1996.
12. R. Braumandl, M. Keidl, A. Kemper, D. Kossmann, A. Kreutz, S. Seltzsam, and K. Stocker, *ObjectGlobe: Ubiquitous query processing on the internet*, The VLDB Journal, vol. 10, no. 1, 2001.
13. F. Naumann and U. Lesser, *Quality–driven integration of heterogeneous information systems*, in Proceeed- ings of the 25th International Conference on Very Large Data Bases (VLDB), Edinburgh, UK, Sept. 1999.
14. J.M. Hellerstein, M.J. Franklin, S. Chnadrasekaran, A. Deshpande, K. Hildrum, S. Madden, V. Ramana, and M.A. Shah, *Adaptive query processing: Technology in evolution*, IEEE Data Engineering Bulletin, vol. 23, no. 2, 2000.
15. Z. Ives, D. Florescu, M. Friedman, A. Levy, and D. Weld, *An adaptive query execution system for data integra- tion*, in Proceedings of the ACM SIGMOD International Conference on Man agement of Data, Philadelphia, PA, USA, June 1999.
16. L. Amsaleg, P. Bonnet, M.J. Franklin, A. Tomasic, and T. Urhan, *Improving responsivness for wide-area data access*, IEEE Data Engineering Bulletin, vol. 20, no. 3, pp. 3–11, 1997.
17. H. Garcia-Molina, Y. Papakonstantinou, D. Quass, A. Rajaraman, Y. Sagiv, J.D. Ullman, V. Vassalos, and J. Widom, *The TSIMMIS approach to mediation: Data models and languages*, Journal of Intelligent Information Systems, vol. 8, no. 2, 1997.
18. A. Levy, A. Rajaraman, and J. Ordille, *Querying heterogeneous information sources using source descrip- tions*, in Proceeedings of the 22nd International Conference on Very Large Data Bases (VLDB), Bombay, India, 1996.
19. O.M. Duschka and M.R. Genesereth, *Query planning in infomaster," in Proceedings of the Twelfth Annual ACM Symposium on Applied Computing*, SAC '97, San Jose, CA, USA, Feb. 1997.
20. D. Florescu, A. Levy, I. Manolescu, and D. Suciu, *Query optimization in the presence of limited access patterns*, in Proceedings ACM SIGMOD International Conference on Management of Data, Philadephia, Pennsylvania, USA, June 1999.

21. Liu, Mengmeng and Ives, Zachary G. and Loo, Boon Thau, *Enabling Incremental Query Re-Optimization*, Proceedings of the 2016 International Conference on Management of Data, 978-1-4503-3531-7.
22. Ramachandra, Karthik and Sudarshan, S., *Holistic Optimization by Prefetching Query Results*, Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, 978-1-4503-1247-9.
23. Moh, Teng-Sheng and Irani, Jehaan, *Random Selection Assisted Long Web Search Query Optimization*, Proceedings of the 50th Annual Southeast Regional Conference, 2012, 978-1-4503-1203-5.
24. Lipton, Richard J. and Naughton, Jeffrey F. and Schneider, Donovan A., *Practical Selectivity Estimation Through Adaptive Sampling*, Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data, 0-89791-365-5.
25. Hellerstein, Joseph M. and Haas, Peter J. and Wang, Helen J., *Online Aggregation*, Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data, 0-89791-911-4.

AUTHORS PROFILE

Dr. Shashidhara H R, has received M.Tech., Degree from Visvesvaraya Technological University, Belagavi, Karnataka in 2003 and Ph.D. from Reva University, Bengaluru, Karnataka in 2018. Currently working as an Associate Professor in the Department of Computer Science & Engineering, RNS Institute of Technology, Bengaluru. He has 27 years of experience in teaching. His area of research interests include Web Mining, KDD, Image Processing, Pattern Recognition. He has published more than 16 papers in leading reputed International Conferences/Journals/National conferences.

Sanjay P K, has received M.Tech., Degree from Visvesvaraya Technological University, Belagavi, Karnataka in 2003. Currently working as an Assistant Professor in the Department of Computer Science & Engineering, RNS Institute of Technology, Bengaluru. He has 13 years of experience in teaching. His area of research interests include Web Mining, KDD, Pattern Recognition. He has published more than 4 papers in leading reputed International Conferences/Journals/National conferences.

Dr. G T Raju, has received M.E. Degree from Bangalore University, in 1995 and Ph.D. from Visvesvaraya Technological University, Belagavi, Karnataka in 2008. Currently working as Vice Principal, Professor and Head, in the Department of Computer Science & Engineering, RNS Institute of Technology, Bengaluru. He has 26 years of experience in teaching and research. His area of research interests include Web Mining, KDD, Image Processing, Pattern Recognition. He has published more than 90 papers in leading reputed International Journals/ Conference proceedings. He has authored five Technical books. He has completed two funded research projects. Thirteen Research Scholars have been awarded Ph.D. Degree under his supervision.

Dr. M Vinayakamurthy, has received M.Sc., in Mathematics from Bharathidas University and Ph.D., in Computational Fluid Dynamics – Mathematics from Bangalore University. He has 26 years of experience in teaching and research. His area of research interest include Data Mining, Data Warehousing, Probability and Statistics, Operation Research and System Simulation & Modeling. He has published more than 20 papers in leading reputed International Journals/ Conference proceedings. He has authored 37 Text books.