International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-4 Issue-2, July 2014

Optimizing Big Data

Anuranjan Misra, Anshul Sharma, Preeti Gulia, Akanksha Bana

Abstract-When sales representatives and customers negotiate, it must be confirmed that the final deals will render a high enough profit for the selling company. Large companies have different methods of doing this, one of which is to run sales simulations. Such simulation systems often need to perform complex calculations over large amounts of data, which in turn requires efficient models and algorithms. This paper intends to evaluate whether it is possible to optimize and extend an existing sales system called PCT, which is currently suffering from unacceptably high running times in its simulation process. This is done through analysis of the current implementation, followed by optimization of its models and development of efficient algorithms. The performance of these optimized and extended models is compared to the existing one in order to evaluate their improvement. The conclusion of this paper is that the simulation process in PCT can indeed be optimized and extended. The optimized models serve as a proof of concept, which shows that results identical to the original system's can be calculated within < 1% of the original running time for the largest customers.
Keywords: - PCT, optimized, algorithms, simulations.

I. INTRODUCTION

Big data is a slightly abstract phrase which describes the relation between data size and data processing speed in a system. A comprehensible definition of the concept is \data whose size forces us to look beyond the tried-and-true methods that are prevalent at that time." [1]. This means that a scenario where innovative optimization of both models and algorithms is required to handle large amounts of data might well be classified as a big data problem. In PCT, the big data challenge arises from the huge amounts of data needed in order to run simulations for large customers. In some cases more than fifty thousand historical order rows may have to be handled, with multiple possible conditions and discount rates applied to every single one of them. While the data set itself is not extremely large by today's standards, the complex operations and calculations which have to be performed on each one of them adds new dimensions to the simulation procedure. Discounts are for example inherited through a large tree structure containing tens of thousands of nodes and the results must be presented to the user within a reasonable amount of time. The reasonable time limit has been defined as ten seconds for the simulation procedure in PCT. This value is based on research [2, 3] showing that a system user who has to wait even further for results of complex calculations will lose focus - something which could prove devastating during a negotiation with a customer.

Manuscript published on 30 July 2014.
*****Correspondence Author(s)
Dr. Anuranjan Misra, Prof. & Dean at Bhagwant Institute of Technology, Ghaziabad, India.
Ms. Anshul Sharma, M.Tech Scholar of Department of Computer Science & Applications Maharshi Dayanand University, Rohtak, India.
Dr. Preeti Gulia, Asst. Prof., Department of Computer Science & Applications, Maharshi Dayanand University, Rohtak, India.
Ms. Akanksha Bana, Asst. Prof. in Computer Science & Engineering Department at Bhagwant Institute of Technology, Ghaziabad, India.

An ideal simulation procedure would always return the results within just a few seconds, since this would mean that simulations could take place during normal conversation without requiring any waiting at all.

A. GOALS AND LIMITATIONS

The first goal of this project is to optimize the existing discount simulation algorithm in order to reduce its running time. The discount simulation's purpose is to apply given discounts to articles and article categories, in order to evaluate whether they will generate an acceptable profit for the selected customer. The second goal is to create a model with associated algorithms for a scaling extension of the system's simulation functionality. The purpose of this extension is to make it possible to apply different discount rates depending on the volume of individual orders.
This will encourage customers to place a few large orders every year instead of several small ones, thus decreasing shipping and warehouse charges for the company without reducing the sales volumes.

B. PAPER OUTLINE

The rest of this report is divided into four chapters - Simulation, Method, Results and Discussion.
The Simulation chapter begins with a detailed description of how discount rate simulations work and the problems which the current implementation has introduced. The second part contains a specification of the scaling simulation functionality and an explanation of the technical difficulties which are introduced by this extension. The Method chapter describes the models and algorithms which have been developed in this project. It also contains a theoretical analysis of these and comparisons between the current implementation in PCT and our solution.
In the Results chapter, the performance of PCT as well as of our solutions for both the optimized customer discount model and the scaling extension are presented. This is split up into a set of test cases, with motivations of their relevance for actual usage scenarios.

II. SIMULATION

When a sales representative negotiates with a customer, one can think of it as a sort of balancing problem. The sales representative wishes to maximize the profit gained by keeping discounts at a minimum, while the customer wants to minimize his or her costs by maximizing the discounts. This is where the simulation process comes in handy -by simulating the effects of new discounts, it is possible to decide whether they are profitable enough or not. When both the sales representative and the customer are satisfied with the results, they can save the discounts as conditions in the system's database. Discount rates from such conditions will then be applied to the customer's future orders.

A. CUSTOMER DISCOUNT SIMULATION

Customer discount simulations are currently fully implemented in PCT. By running a simulation over the data described in section 2.1.1, a sales representative will find out which profit would be gained if the customer bought the same articles as in the historical period but using current pricing conditions. Even more importantly, new discount rates can be applied to the simulation meaning that the sales representative can see which effects they will give and whether they seem profitable enough or not. The details of the simulation process are described first in section 2.1.2, but reading the chapter in the presented order is highly recommended. Understanding of the underlying concepts is a great advantage when trying to gain insight into the workings of the simulation process.

A. 1 DATA NEEDED FOR A CUSTOMER DISCOUNT SIMULATION

A simulation is based on data from the following sources:

a) Article tree - A tree structure where branch nodes represent article categories and leaf nodes represent articles
b) Sales history - A set of aggregated order rows, containing information about previous sales history
c) Existing customer conditions - Agreed discount rates from existing contracts, which set a certain discount rate to a specific node in the article tree
d) User input - Various parameters that specify which historical data and discount rates to use in the simulation.

A. 2 THE SIMULATION PROCESS

The sales representative starts by entering which customer he is negotiating with and selecting a path in the article tree for which discounts will be entered. Next up, a start and stop month is specified and now the system is ready to run the first simulation. Since no discount rates have been entered at this point, all nodes in the path will use their existing discount rates if any such exist in the active conditions and 0:0% otherwise. All price level 1 nodes which are not affected by the existing conditions will also have their discounts set to 0:0%. Due to the concept of discount inheritance, all other nodes will inherit their parent's discount rate top-down if they do not have an existing condition. This means that the results of the first run will always show the economical results that will follow if the same item quantities are sold as in the historical data used for the simulation, taking only currently active conditions into account. Conditions may have been added or removed since the historical orders were handled, so it is not enough to just aggregate the values and profits from the history database. Instead, the \base value" (which one can think of as the price for the order rows if no discounts had been applied) must be calculated for each article. By applying discount rates from existing conditions to these base values, the system finds out how much the customer would have to pay for the same orders if they had been placed using current conditions. In the next step, the sales representative sets discounts for the nodes in the selected path and runs another simulation over the same data. Any conditions affecting discount rates for the path nodes will be overrun by the discount rates set by the sales representative, while conditions affecting other nodes will still be taken into consideration. The user specified discount rates will then be inherited down through the article tree just like the ones from the conditions. The result will thereby correspond to the profit which would be achieved if these new rates were added to the conditions database and the same orders as in the historical data were then placed again by the customer. This simulation step will typically be run multiple times with different discount rates for the nodes in the path, until they are balanced in such a way that both the customer and the sales representative are satisfied with the results. Running multiple simulations with different discount rates for the same time period and historical data until one gets satisfying results is referred to as going through a simulation process.

III. SIMULATION OUTPUT

So far, the output of simulations has been described in terms of \profit" and \value". The actual values computed during a simulation are of course more specific than that and as such, the specification of requirements presents guidelines for the output data layout. The specification indicates that the output should be presented as a table, where each node in the selected path is represented as a row. There is also a top row labeled \Total", which shows the total simulation values of all articles in the whole article tree. A print screen showing how this looks in the current version of PCT is shown in figure 2.1. The columns of each row are described in table 2.3.

Customer Total	Volume (kg)	Value (EURO)	C0	C0%	Actual Discount	Agreed Discount
Total	128,167	471,233	365,257	77.5	86.4	

Price Level 1	Volume (kg)	Value	C0	C0%	Actual Discount	Agreed Discount	Avg. Agreed	Target	Avg. Target
PL1_10	114,635	446,862	355,066	79.5	87.1	44.0 * 0.0	0.0	58.5	66.2

Price Level 2	Volume (kg)	Value	C0	C0%	Actual Discount	Agreed Discount	Avg. Agreed	Target	Avg. Target
PL2_01	10,349	24,014	15,670	65.3	87.6	67.1 * 0.0	0.0	66.9	66.9

Price Level 3	Volume (kg)	Value	C0	C0%	Actual Discount	Agreed Discount	Avg. Agreed	Target	Avg. Target
PL3_5751F1	26	187	161	86.0	78.1	12.4 * 0.0	0.0	54.7	54.7

Figure 2.1. A Print Screen from PCT Showing how Simulation Output is Presented in the Current System

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Discount Inheritance

Discounts can be applied to nodes on any level of the article tree - from price level 1 down to specific articles. It is intuitive that a discount which is set for a single article will only affect the price of that specific article. When it comes to discounts set on article groups or price level nodes, the system uses a concept called \discount inheritance" to let this affect underlying nodes. In order to determine which discount rate to apply to a given node, the method presented in algorithm 2.1.1 is used.

Algorithm 2.1.1: Find Discount Rate (Node n)
Input: A node n from the article tree
Result: The discount rate which should be applied to n
1 if n is a node in the path for 1 which a discount rate d is set then
2 return d
3 else if n is not a node in the path AND n has an active condition c then
4 return the discount rate from condition c
5 else if n is a price level 1 node then
6 return 0:0%
7 else
8 parent := n's parent node in the article tree
9 return findDiscountRate(parent)
10 end

The concept of discount inheritance is easy to visualize due to the tree structure of the article database. An example tree with some existing discount rates is shown in figure 2.2. Existing discount rates are written directly onto the grey nodes to which they belong, while nodes without such rates are white. The final result of the discount rate inheritance in the same tree can be seen in figure 2.3, where arrows show how discount rates are passed down through the tree.

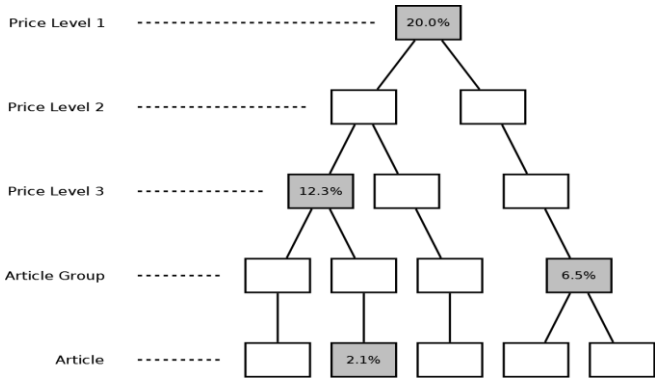

Figure 2.2. An Example Article Tree Where Discount Rates Have Been Set for Four Nodes

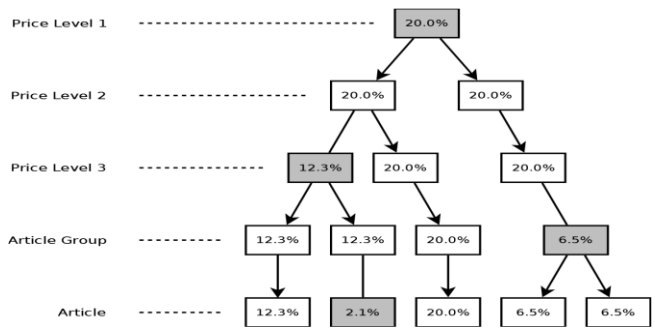

Figure 2.3. Discount Inheritance in the Example Article Tree from Figure 2.2

A. 3 CURRENT IMPLEMENTATION

As mentioned in the project motivation in section 1.1, the current implementation of PCT suffers from critical performance issues. Since the source code of this system is not allowed to be included in this report, the problems of its algorithm have to be explained in terms of bad structure choices and complexity rather than examples and excerpts from the actual code. A (very) rough outline of the algorithm structure used to perform simulations in PCT is presented in algorithm 2.1.2. While it does not motivate or explain the details of each step, it does provide enough information to analyze its complexity. To give the reader some sort of idea of the actual magnitude of the implementation of this algorithm, its Java source code takes up several hundred kilobytes (not including GUI, server connections, database handling and other parts which are not directly related to the algorithm). In other words, a line describing e.g. criteria matching means running a separate algorithm which in turn has a complexity worth mentioning.

Algorithm 2.1.2: Structure of the simulation process in PCT
1 if this is the first 1 run of the simulation process then
2 initialize connection to each input data element in the GUI [O(k)]
3 end
4 for each price level in the article tree [O(k)] do
5 match condition level [O(k)]
6 match price level [O(k)]
7 for each item in the customer's cache [O(n)] do
8 match criteria [O(k)]
9 end
10 retrieve target discount [O(k)]
11 for each article in the article tree [O(a)] do
12 foreach article in the customer's cache [O(n)] do
13 match criteria [O(k)]
14 foreach price level in the article tree [O(k)] do
15 retrieve data and calculate results
16 end
17 end
18 retrieve agreed discounts [O(k)]
19 compare discounts to target discounts [O(k)]
20 end
21 end
22 for each article in the customer's cache [O(n)] do
23 calculate results for articles under price level 1 nodes 2= path
24 end

In the pseudo code above, the complexity has been included on each line where O notation is applicable. The meaning of each occurring variable in the O notation is presented.

The total complexity of the implementation of the current simulation algorithm is

$O(k+k(k+k+nk+k+a(n(k+k))+k+k)+n) = O(k+5k2+nk2+2ank2+n) = O(ank2)$

It should also be noted that the complexity of repeated runs of the algorithm is

$O(k(k+k+nk+k+a(n(k+k))+k+k)+n) = O(5k2+nk2+2ank2+n) = O(ank2)$.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

IV. RESULTS

This section contains running times of customer discount simulations. Running times for our implementation are

shown together with corresponding running times for PCT for the same underlying data.

Articles	Running Time PCT [ms]	Running Time Our Model [ms]
1	723	130
10	876	156
40	741	89
105	1,142	91
206	1,879	118
366	4,671	131
483	6,473	148
789	9,141	161

Table 4.1. Running Time for First Simulation in PCT and Our Model

Running time [ms] #articles	PCT Run1 Run2 Run 3	Our model Run1 Run2 Run 3
40	741 782 692	89 < 1 < 1
206	1,879 1,801 1,707	118 < 1 < 1
366	4,671 3,879 4,665	131 < 1 < 1
483	6,473 6,773 4,240	148 < 1 < 1
789	9,141 6,780 6,087	161 < 1 < 1

Table 4.2. Running Time for Repeated Simulations in PCT and Our Model

articles	Running time PCT [ms]	Running time our model [ms]
100	736	150
500	1,295	161
1,000	1,694	157
1,500	1,835	157
2,000	2,161	153
3,000	2,884	150
5,000	4,335	152
10,000	8,463	146
20,000	20,314	166
30,000	33,671	210
40,000	45,892	253

Table 4.3: Running Time for First Simulation in PCT and Our Model Over Generated Data

V. FUTURE WORK

Another interesting approach would be a comparison between the performances of our models using different database solutions. NoSQL database systems could prove effective in handling the big data problems introduced by the scaling extension. Particularly, an implementation using a graph database would be interesting due to this technology's great performance when dealing with tree structures. For example, the graph database Neo4j has shown promising results in multiple studies such as [6], where Neo4j is concluded to be up to ten times faster than MySQL for traversals and [7], where the results show that running times for MySQL increase much faster than for Neo4j as the data magnitude grows.

VI. CONCLUSIONS

This project has consisted of analysis, optimization and implementation of the existing simulation algorithms in PCT as well as modeling and implementation of its upcoming scaling extension. The results show that the implementation of the optimized customer discount model provides large enough performance improvements to guarantee reasonable running times even for the largest customers. The results for the scaling extension prove that implementation of the desired functionality in PCT is possible as well, as long as the big data issue is handled in an efficient way. A final conclusion of this project is that optimization of existing algorithms is not always sufficient in order to improve the performance of a system. Creating new, optimized models and developing fast algorithms for these can prove far more efficient than optimization of existing algorithms based on improficient models.

REFERENCES

1. A. Jacobs, The pathologies of big data, Commun. ACM Vol. 52 (8) (2009) pp. 36{44.
2. R. B. Miller, Response time in man-computer conversational transactions, in: Proceedings of the December 9-11, 1968, fall joint computer conference, part I, AFIPS '68 (Fall, part I), New York, NY, USA, 1968, pp. 267{277.
3. S. C. Seow, User and system response times, in: Designing and Engineering Time: The Psychology of Time Perception in Software, Addison-Wesley Professional, 2008, pp. 33{48.
4. M. de Berg, O. Cheong, M. van Kreveld, M. Overmars, 1-dimensional range searching, in: Computational Geometry: Algorithms and Application 2ed, Springer Berlin Heidelberg, 2008, pp. 96{99.
5. Y. Manolopoulos, Y. Theodoridis, V. J. Tsotras, Advanced Database Indexing, Springer US, 2000, Ch. 3. Fundamental Access Methods, pp. 37{59.
6. C. Vicknair, M. Macias, Z. Zhao, X. Nan, Y. Chen, D. Wilkins, A comparison of a graph database and a relational database: a data provenance perspective, in: Proceedings of the 48th Annual Southeast Regional Conference, ACM SE '10, ACM, New York, NY, USA, 2010, pp. 42:1{42:6.
7. S. Batra, C. Tyagi, Comparative analysis of relational and graph databases, International Journal of Soft Computing and Engineering (IJSCE) Volume 2 (Issue 2) (2012) pp. 509{512.

AUTHOR PROFILE

Prof. (Dr.) Anuranjan Misra, is professor & Dean at Bhagwant Institute of Technology, Ghaziabad, India. He had authored 30 books, 100 research papers. His books are in many Indian & foreign universities syllabus.

Ms. Anshul Sharma, is M.Tech Scholar of Department of Computer Science & Applications Maharshi Dayanand University, Rohtak, India.

Dr. Preeti Gulia, is an Assistant Professor, Department of Computer Science & Applications, Maharshi Dayanand University, Rohtak, India.

Ms. Akanksha Bana, is an assistant professor in computer science & engineering department at bhagwant institute of Technology, Ghaziabad, India.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)