MASTER BUSINESS MATHEMATICS AND INFORMATICS THESIS BY MARTINA HOEVER

PERFORMANCE ANALYSIS OF AN INFORMATION FACTORY

A DISCRETE EVENT SIMULATION MODEL USING SHARED PROCESSES

By Martina Hoever
Student ID 1706144
Employee ID 354864
Period February 2013 – September 2013
Degree Master Business Mathematics and Informatics / Business Analytics
Tutor Cognizant Technology Solutions Wiljo Verbruggen
Tutor Vrije Universiteit Rob van der Mei, Patricia Lago

ABSTRACT

In the summer of 2012, Philips signed a multi-year outsourcing IT contract with Cognizant Technology Solutions. One of Philips' questions to Cognizant concerns the means to accomplish the following two goals: "To treat information as a differentiating asset to Philips and make that information available as a service to all those who are in need of timely, complete and accurate data." This study focuses on the performance of this Information Factory, specifically the near real-time requirement. This performance is measured by the ability of the systems to get all the data in the Information Factory in a timely manner.

The conclusions of this research are:

- Determining the bottlenecks and increasing their capacity results in a very large decrease of transaction times.
- For realistic load scenarios, increasing the capacity of the ESB to 8 million bytes per second and the capacity of the users to 240 million bytes per second shows an average transaction time of 5 seconds.
- Properly increasing the capacity of only two components resulted in a reduction in transaction time by a factor of 10.

PREFACE

This thesis is the result of individual research performed on a practical problem during my six-month internship at Cognizant Technology Solutions. This internship is an important part of the Business Mathematics and Informatics master program at the Vrije Universiteit in Amsterdam. The goal of is to encompass the program components business, mathematics, and informatics in practice at an external business, industry or research facility.

The present report contains the results of my internship done at Philips as a business analyst for Cognizant Technology Solutions. In this report I will analyze the performance of the Information Factory built for Philips. I will focus on the ability of extracting data real-time.

I would like to thank Wiljo Verbruggen for giving me the opportunity to undertake an internship at Cognizant. Furthermore, I would like to thank Pierre Puts and Sjaak L'Ami for guiding me in my time at Philips. I also would like to thank Rob Konterman for introducing me to the Cognizant.

I would like to thank the Vrije Universiteit for offering the possibility of writing your master thesis at a firm, so you can get experience while still studying. This thesis would not have been possible without the help, support of my advisor, Rob van der Mei, whom I want to thank for his excellent guidance. Next to this, I would like to thank my second reader Patricia Lago. Furthermore, I would like to thank Annemieke van Goor for consulting me during my master.

Amsterdam, September 2013

MANAGEMENT SUMMARY

Context

In the summer of 2012, Philips signed a multi-year outsourcing IT contract with Cognizant Technology Solutions. One of Philips' questions to Cognizant concerns the means to accomplish the following two goals: To treat information as a differentiating asset to Philips and make that information available as a service to all those who are in need of timely, complete and accurate data. To accomplish these goals, an Information Factory will be created. An Information Factory is a software component that acts as a data service, i.e. it makes available one or more service operations through which service consumers can access information present in the factory. In simple terms, an Information Factory combines a data store with a service interface. The Information Factory has the following requirements:

- The Information Factory should be near real-time: A change in the data in the source systems should be visible in the Information Factory within 8 minutes.
- There should be minimal to no footprint on the SAP R/3 side to maintain the adaptability and speed of rollout.
- The SOA-enabled aspect of the Information Factory, which means that direct writes into the Teradata database should be avoided.
- The Information Factory should be data model agnostic: it should be capable of adopting a chosen data model through a domain model.

Goal

To be able to meet all these requirements, the relation between capacity and performance needs to be clear. This study focuses on the performance of this Information Factory, specifically the near real-time requirement. This is measured by the ability to get all the data in the Information Factory in a timely manner. The goal is to be able to predict, for different capacity estimations, if the system is able to meet the near real-time requirement.

Methods

To meet the project goal, it is key to determine the size and distribution of data that will be the input for the system. Next to this, it must be determined how much data the different components of the system can handle. Only when all this is known, an accurate performance measurement can be done. However, the size and distribution of the input data, as well as the amount of data the system can handle, can change strongly over time. A prediction for future requirements can be determined by creating a simulation of the behavior of the different systems when under different data demands. This should result in a simulation model of the availability of the Information Factory, where the delay in each system is calculated, with different capacity options as input. This analysis should be repeatable, with different parameters or renewed historical data.

Results and conclusions

The Enterprise Service Bus is the bottleneck under the current estimations. Even though the simulation showed that data was able to transfer to the output components within the near-real time requirement, it is likely that this won't be the case in the long run. If the four queues that are increasing keep increasing with the same rate, the near-real time requirement will not be met while running the system for a longer period.

The different kernels have different amounts of data to be transferred, which could result an average transfer rate meeting the near real time requirement, while some of the kernels might not meet this requirement. When the system is in place, it is important that a check takes place whether *all* kernels meet the requirement, and not only the total average.

The way data is transferred after the PowerCenter component can have a large influence on the transaction time. Even though direct writes to the Teradata database are to be avoided, the high increase in speed suggests that this option needs to be re-evaluated.

Determining the bottlenecks and increasing their capacity results in a very large decrease in transaction times. Increasing the capacity of the ESB and the users shows an average transaction time of 5 seconds. Changing the capacity of only two components resulted in a reduction of transaction time by a factor of 10.

TABLE OF CONTENTS

CONFIDENTIAL

1 BIBLIOGRAPHY

Balsamo, S. (2003). A review on queueing network models with finite capacity queues for software architectures performance prediction. *Performance Evaluation*, 269-288.

Borst, S. (n.d.). *The equivalance between processor sharing and service in random order.* Amsterdam: Centrum voor Wiskunde en Informatica.

Boukachabine, A. (2013). *Oracle Database: The Database of Choice for Deploying SAP solutions.* Redwood Shores: Oracle.

Cheng, L. (2004). Logistics for orld-wide crude oil transportation using discrete event simulation and optimal control. *Computers & Chemical Engineering*, 897-911.

Coffman, E. G. (1970). Waiting Time Distributions for Processor-Sharing Systems. *Journal of the Association for Computing Machinery*, 123-130.

Cohen, J. W. (1979). The Multiple Phase Service Network with Generalized Processor Sharing. *Acta Informatica*, 245-284.

Dallery, Y. (1992). Manufacturing flow line systems: a review of models and analytical results. *Queueing Systems*, 3-94.

Faget, P. (2005). *Applying Discrete Event Simulation and an Automated Bottleneck Analyis as an Aid to Detect Running Production Constraints.* Gothenburg: Chalmers University of Technology.

Fujimoto, R. M. (1990). Parallel Discrete Event Simulation. *Communications of the ACM*, 30-53.

Günal, M. M. (2010). discrete event simulation for performance modelling in health care: a review of the literature. *Journal of Simulation*, 42-51.

Helbing, D. (2003). A section-based queueing-theoretical traffic model for congestion and travel time analysis in networks. *Journal of Physics A: Mathematical and General*.

Hooyman, B. (2012). *Platform vison for the enabling platform - Information.* Eindhoven: Philips.

Hung, G. R. (2007). Computer Modeling of Patient Flow in a Pedriatric Emergency Department Using Descrete Event Simulation. *Pediatric Emergency Care*.

Jie-Zhang, L. (2011). *Innovations, Standards, and Practices of Web Services.* IGI Global.

K. Parekh, A. (1993). A Generalized Processor Sharing approach to Flow Control in Intergated Services Networks: The Single-Node Case. *IEEE Transactions on Networking*.

Koole, G. (2010). *Optimization of Business Processes: An Introduction to Applied Stochastic Modeling.* Amsterdam: VU Universty Amsterdam.

L'Ami, S. (2013). *End2End Performance & Capacity Management Information Factory.* Eindhoven.

Nelson, D. (2004). *The Penguin Dictionairy of Statistics.* London: Penguin Books.

Nicol, D. M. (1988). *Parallel Discrete-Event Simulation of FCFS Stochastic Queuing Networks.* Williamsburg: Department of Computer Science of The College of William and Mary.

O'Donovan, T. M. (1974). Direct Solutions of M/G/1 Processor Sharing Models . *Operations Research*, 1232-1235.

Ott, T. J. (1984). The Sojoutn-Time Distribution in the M/G/1 Queue with Processor Sharing. *Journal of Applied Probability*.

Pavlos Konas, P.-C. Y. (n.d.). *Parallel Discrete Event Simulation on Shared-Memory Multiprocessors.* Urbana: Center for Supercomputing Research and Development.

Perekh, A. K. (1994). A Generalized Processr Sharing Approach to Flow Control in Integrated Services Networks: The Multiple Node Case. *IEEE Transactions on Networking*.

Puts, P. (2013). *Connecting SAP R/3 to the Information Factory.* Eindhoven: Cognizant.

Reed, D. A. (1988). *Parallel Discrete Event Simulation Using Shared Memory.* IEEE Transactions On Software Engineering.

Reiser, M. (1979). A Queueing NetworkAnalysis of Computer Communication Networks with Window Flow Control. *IEEE Transactions on Communications*.

Robinson, S. (1994). *Succesful simulation: a practical approach to simulation projects.*

Schassberger, R. (1984). A New approach to the M/G/1 Processor-Sharing Queue. *Advances in Applied Probability*, 202-213.

Siebers, P. O. (2010). Discrete-event simulation is dead, long live agent-based simulation! *Journal of Simulation*, 204-210.

Singh, V. (2013). *Notes on the Informatica Grid Discussion.*

Swisher, J. R. (2003). Discrete-Event Simulation Optimzation Using Ranking, Selection, and Multiple Comparison Procedures: A Survey. *ACM Transactions on Modeling and Computer Simulation*, 134-154.

Thiruvengadam, B. (2013). *Philips Information Factory service specification document.* Cognizant.

Tijms, H. C. (1986). *Stochastic Modeling and Analysis: a Computational Approach.* Amsterdam: John Wiley & Sons.

Uitert, M. J. (2003). *Generalized Processor Sharing Queues.* Amsterdam: Centrum voor Wiskunde en Informatica.

Wu, D. (2003). Effective Capacity: A Wireless Link Model for Support of Quality of Service. *IEE Transactions on Wireless Communictions*.

Zhang, Z.-L. (1995). Statistical Analysis of the Generalized Processor Sharing Scheduling Discipline. *IEE Journal on Selected Areas in Communications*.

CONFIDENTIAL