QUEUED TRANSACTION PROCESSING WITH WEB SERVICE RELIABLE MESSAGING

A.C. SURIARACHCHI

This Dissertation was submitted to the Department of Computer Science and Engineering of the University of Moratuwa in partial fulfillment of the requirements for the Degree of MSc in Computer Science specializing in Software Architecture

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Department of Computer Science and Engineering
University of Moratuwa
February 201 0

96424

ABSTRACT

With the popularity of the distributed business applications, the application data is distributed in various physical storages. However most of the business transactions require to update data stored in more than one storage. hence updating two data storages reliably is a common problem for most of the distributed business applications.

Queued transaction processing is a concept widely used to achieve such a processing model using intermediate queues to transfer messages reliably. In such a system at the client side, both updating the client storage and writing the message to be sent to the client side message queue happens in the same distributed transaction. Similarly at the server side reading the message from the server side queue and updating the sever storage happens in the same distributed transaction. Bur such a system may have interoperability problems if client and server use different types of technologies.

Web services are used to communicate among the heterogeneous systems by passing SOAP messages using standard transport mechanisms like http. Web services can reliably communicate by using WS-Reliable messaging specification(WS-RM). WS-RM uses concepts of Reliable messaging source (RMS) and Reliable messaging destination (RMD) between which it guarantees reliable massage delivery.

By combining these two concepts, we introduce an approach to solve the above mentioned problem in an interoperable manner using WS-RM ..,to communicate between nodes while keeping RMS and RMD as intermediate storages. In our model reliable message delivery happens in three phases. First both updating application client storage and writing message to the RMS happens in the same distributed transaction. Then WS-RM protocol reliably transfers the message to RMD at the server side . Finally- at the server reading the message from the RMD and updating the server storage happens in the same distributed transaction. The middleware software entity that we developed to encapsulate this approach is called Mercury which implements WS-RM protocol.

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

DECLARATION

"The work included in this report was done by me, and only by me, and the work has not been submitted for any other academic qualification at any institution"

[illegible]

Name: A.C. Suriarachchi (088254P)
Date: 2010.02.26

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

"I certify that the declaration above by the candidate is true to the best of my knowledge and that this dissertation is acceptable for evaluation for the Degree of M.Sc in Computer Science specializing in Software Architecture"

UOM Verified Signature

Project Supervisor: Dr. Sanjiva Weerawarana
Date: 2010.02.26

ACKNOWLEDGMENTS

I wish to sincerely thank my supervisors Mr. Paul Fremantle and Dr. Sanjiva Weerawarana for providing me the research idea and supervision of my work continuously. They provided me necessary guidance, various levels of requirements and encouragement to fulfill my objective. I would also like to thank Dr Srinath Perera who reviewed my work and provided me valuable feedback. I am also grateful to Prof. Gihan Dias and Dr Sanath Jayasena who worked as the course coordinators, and provided valuable feed back at various levels of the project. I would like to extend my thank to all the academic staff of the University of Moratuwa for the great work they did for us during the course of study.

No student can survive in a university without the help of their fellow students to discuss ideas, share opinions, and to make time spent in the lab and all round enjoyable experience. I would be grateful for all M.Sc 08 colleagues for the corporation given to the successful completion of my project involvements.

I must also be grateful to my parents and brothers for the encouragement they provided to follow the Msc. I wish to express my gratitude to all my colleagues at WSO2 who have enormously helped to learn a lot about web services and distributed systems.

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

TABLE OF CONTENTS

DECLARATION i
ABSTRACT ii
ACKNOWLEDGMENTS iii
TABLE OF CONTENTS iv
LIST OF FIGURES vi
LIST OF ABBREVIATIONS viii
Chapter 1 Introduction 1
1.1 Background 1
1.2 Abstract Problem 2
1.3 Method of study 3
1.4 Previous work 3
1.5 Expected result 3
Chapter 2 Literature Review 4
2.1 Web service standards 5
2.1.1 SOAP 5
2.1.2 WS-Addressing 5
2.1.3 WS-Reliable messaging 6
2.2 Transactions and messaging standards 9
2.2.1 2PC 9
2.2.2 X/Open distributed transaction standards 10
2.2.3 JTA 11
2.3 WS-Transactions 13
2.3.1 WS-Coordination 13
2.3.2 WS-Atomic transactions 15
2.3.3 WS-BussinessActivity 15
2.4 Queued Transaction processing 16
2.5 Different types of reliable web services 16
2.5.1 Using Message Oriented Middleware for Reliable Web Services Messaging. 16
2.5.2 Three facets of Reliability 17
2.5.3 Reliable messaging for web services 17
2.5.4 Assessment 18
2.6 Transactions and messaging 20
2.6.1 MQ Integrating Transactions 20
2.6.2 Message delivery transactions 20
2.6.3 Message processing transactions 21
2.6.4 Full messaging transactions 22

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

2.7 Middleware mediated transactions 22

2.7.1 D-sphere 23

Chapter 3 Methodology 25

3.1 Previous Solution 25

3.2 Proposed Solution 26

3.3 Alternative Solutions 27

3.3.1 Integrate WS-RM protocol with the client storage and server storage by taking them as RMS and RMD 27

3.3.2 Use the same transaction to update both application level storages and WS-RM storages 28

3.4 Solution Architecture 29

3.4.1 State machine model 29

3.4.2 Run time Architecture 36

3.4.3 Storage API 40

3.4.4 Other issues and solutions 45

Chapter 4 Use case scenarios 46

4.1 InMemory 47

4.1.1 In Only invocation 47

4.1.2 In Out Invocation 48

4.1.3 Fault Handling 49

4.2 Persistence 49

4.2.1 In Only Invocation 50

4.2.2 In Out Invocation and Fault Handling 51

4.3 JTA 52

4.3.1 In Only Invocation 55

Chapter 5 Observations & Results 57

Chapter 6 Conclusion & Future Work 59

REFERENCES 61

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

LIST OF FIGURES

Figure 1-1 Abstract Problem 2
Figure 2-1 Reliable Messaging Model 7
Figure 2-2 Reliable Messaging Protocol 7
Figure 2-3 Coordinator States 9
Figure 2-4 Participant States 10
Figure 2-5 X/Open Distributed Transaction Standards 11
Figure 2-6 JTA Overview 12
Figure 2-7 WS Coordination Framework 14
Figure 2-8 Queued Transaction Processing 16
Figure 2-9 MQ Integrating Transactions 20
Figure 2-10 Message Delivery Transactions 21
Figure 2-11 Message Processing Transactions 22
Figure 2-12 Application Without D-sphere 23
Figure 2-13 Application With D-sphere 23
Figure 2-14 D-sphere Architecture 24
Figure 3-1 Previous Solution 25
Figure 3-2 Proposed Solution 26
Figure 3-3 Integrate Persistence Storage with RM Storages 27
Figure 3-4 Using the same Transaction 28
Figure 3-5 RMSSequence Events 29
Figure 3-6 RMSSequence States 30
Figure 3-7 RMSSequence State Machine 31
Figure 3-8 RMDSequence Events 31
Figure 3-9 RMDSequence States 32
Figure 3-10 RMDSequence State Machine 33
Figure 3-11 InvokerBuffer Events 34
Figure 3-12 InvokerBuffer States 35
Figure 3-13 InvokerBuffer State Machine 35
Figure 3-14 InOnly Messages Runtime 36
Figure 3-15 InOut Messages Runtime 38
Figure 3-16 Fault Handling Runtime 39
Figure 3-17 Storage API 40
Figure 3-18 InMemory Implementation 41
Figure 3-19 Persistence Implementation 42
Figure 3-20 Database Design 44
Figure 4-1 InMemory Invocation 47

UNIVERSITY OF MORATUWA, SRI LANKA LIBRARY

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Figure 4-2 In Only Client 47
Figure 4-3 InMemory In Out Client 48
Figure 4-4 SendAsynchornousMessage Method 48
Figure 4-5 Fault Message Receiver 49
Figure 4-6 Using Persistence Storage 49
Figure 4-7 Persistence Storage Configuration 50
Figure 4-8 Persistence In Only Client 50
Figure 4-9 Persistence In Only MessageReceiver 51
Figure 4-10 Persistence Callback Handler Methods 52
Figure 4-11 JTA Invocation 52
Figure 4-12 JTA Storage Configuration 53
Figure 4-13 Atomikos JTA Connection Manager 54
Figure 4-14 JTA In Only Client 55
Figure 4-15 InvokerWorker 56

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

LIST OF ABBREVIATIONS

WS	Web Service
WS-RM	Web Service Reliable Messaging
RMS	Reliable Messaging Source
RMD	Reliable Messaging Destination
2PC	Two Phase Commit
JTA	Java Transaction API
JTS	Java Transaction Service
SOAP	Simple Object Access Protocol
RPC	Remote Procedure Calls
MOM	Message Oriented Middleware

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk