



Rafael Oliveira Vasconcelos

**A Dynamic Load Balancing Mechanism for Data Stream
Processing on DDS Systems**

Dissertação de Mestrado

Dissertation presented to the Programa de Pós-Graduação em Informática of the Departamento de Informática, PUC-Rio as partial fulfillment of the requirements for the degree of Mestre em Informática.

Advisor: Prof. Markus Endler

Rio de Janeiro
January 9th, 2013



Rafael Oliveira Vasconcelos

**A Dynamic Load Balancing Mechanism for Data Stream
Processing on DDS Systems**

Dissertation presented to the Programa de Pós-Graduação em Informática of the Departamento de Informática, PUC-Rio as partial fulfillment of the requirements for the degree of Mestre em Informática. Approved by the following commission.

Prof. Markus Endler

Advisor

Departamento de Informática – PUC-Rio

Prof. Noemi Rodriguez

Departamento de Informática – PUC-Rio

Prof. Renato Fontoura de Gusmão Cerqueira

Departamento de Informática – PUC-Rio

Prof. José Eugenio Leal

Coordinator of the Centro Técnico Científico – PUC-Rio

Rio de Janeiro, January 9th, 2013

All rights reserved.

Rafael Oliveira Vasconcelos

BSc. in Computer Science at UNIT (Universidade Tiradentes) – 2010.

Ficha Catalográfica

Vasconcelos, Rafael Oliveira

A dynamic load balancing mechanism for data stream: processing on DDS systems / Rafael Oliveira Vasconcelos ; advisor: Markus Endler. – 2013.
74f ; 30 cm

Dissertação (mestrado)—Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2013.
Inclui bibliografia

1. Informática – Teses. 2. Balanceamento de carga. 3. Processamento de fluxo de dados. 4. DDS. 5. SDDL. 6. Publish/Subscribe. 7. Computação autônoma. I. Endler, Markus. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Acknowledgments

First and foremost , I have to thank my parents for all unconditional and eternal support, which includes not only financial support but also loving and educational ones. They are the pillars that hold me up.

Apart from the efforts of myself, the success of any thesis depends largely on the encouragement and guidelines of many others. I take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project. I would like to show my greatest appreciation to my advisor Markus Endler and all my LAC friends. I can't say thank you enough for their tremendous support and help. Without their encouragement and guidance this project would not have materialized. The guidance and support received was vital for the success of this work. I am grateful for their constant support and help.

I also would like to extend my sincere thanks to Twin Oaks Computing, in special to Nina Tucker, for the support provided during this journey.

Lastly, I am grateful for all other people that helped me direct or indirectly.

Abstract

Vasconcelos, Rafael Oliveira; Endler, Markus. **A Dynamic Load Balancing Mechanism for Data Stream Processing on DDS Systems**. Rio de Janeiro, 2013. 74p. MSc. Dissertation – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

This thesis presents the Data Processing Slice Load Balancing solution to enable dynamic load balancing of Data Stream Processing on DDS-based systems (Data Distribution Service). A large number of applications require continuous and timely processing of high-volume of data originated from many distributed sources, such as network monitoring, traffic engineering systems, intelligent routing of cars in metropolitan areas, sensor networks, telecommunication systems, financial applications and meteorology. The key concept of the proposed solution is the Data Processing Slice (DPS), which is the basic unit of data processing load of server nodes in a DDS Domain. The Data Processing Slice Load Balancing solution consists of a load balancer, which is responsible for monitoring the current load of a set of homogenous data processing nodes and when a load unbalance is detected, it coordinates the actions to redistribute some data processing slices among the processing nodes in a secure way. Experiments with large data stream have demonstrated the low overhead, good performance and the reliability of the proposed solution.

Keywords

Load Balancing; Data Stream Processing; DDS; SDDL; Publish/Subscribe; Autonomic Computing.

Resumo

Vasconcelos, Rafael Oliveira; Endler, Markus. **Um Mecanismo de Balanceamento de Carga Dinâmico para Processamento de Fluxo de Dados em Sistemas DDS**. Rio de Janeiro, 2013. 74p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Esta dissertação apresenta a solução de balanceamento de carga baseada em fatias de processamento de dados (Data Processing Slice Load Balancing solution) para permitir o balanceamento de carga dinâmico do processamento de fluxos de dados em sistemas baseados em DDS (Data Distribution Service). Um grande número de aplicações requer o processamento contínuo de alto volume de dados oriundos de várias fontes distribuídas., tais como monitoramento de rede, sistemas de engenharia de tráfego, roteamento inteligente de carros em áreas metropolitanas, redes de sensores, sistemas de telecomunicações, aplicações financeiras e meteorologia. Conceito chave da solução proposta é o Data Processing Slice, o qual é a unidade básica da carga de processamento dos dados dos nós servidores em um domínio DDS. A solução consiste de um nó balanceador, o qual é responsável por monitorar a carga atual de um conjunto de nós processadores homogêneos e quando um desbalanceamento de carga é detectado, coordenar ações para redistribuir entre os nós processadores algumas fatias de carga de trabalho de forma segura. Experimentos feitos com grandes fluxos de dados que demonstram a baixa sobrecarga, o bom desempenho e a confiabilidade da solução apresentada.

Palavras-chave

Balanceamento de carga; processamento de fluxo de dados; DDS; SDDL; Publish/Subscribe; computação autônoma.

Contents

| | | |
|-------|--|----|
| 1 | Introduction | 12 |
| 1.1 | Motivation | 12 |
| 1.2 | Problem Statement | 13 |
| 1.3 | Objective and Contributions | 15 |
| 1.4 | Organization | 16 |
| 2 | Background | 17 |
| 2.1 | DDS | 17 |
| 2.2 | SDDL | 20 |
| 2.3 | Load Balancing in Middleware | 21 |
| 2.3.1 | Classification of Load Balancing Algorithms | 21 |
| 2.3.2 | Elements of a Dynamic Load Balancing Algorithm | 23 |
| 2.3.3 | Virtual Servers | 24 |
| 2.4 | Autonomic Computing | 24 |
| 2.4.1 | MAPE-K Loop | 26 |
| 3 | Data Processing Slice Load Balancing | 28 |
| 3.1 | Data Processing Slice | 30 |
| 3.2 | Assignment Function | 33 |
| 3.3 | DPS Solution | 34 |
| 3.4 | Load Balancing Process | 36 |
| 3.5 | Discussion | 38 |
| 4 | Implementation | 39 |
| 4.1 | MAPE-SDDL | 40 |

| | | |
|-------|--|----|
| 4.2 | Load Balancer | 41 |
| 4.3 | Processing Node | 44 |
| 4.3.1 | MS and LES | 46 |
| 4.3.2 | CES and Load Balancing Process | 47 |
| 4.4 | Load Balancing Algorithms | 49 |
| 5 | Evaluation | 51 |
| 5.1 | Experimental Setup | 53 |
| 5.2 | Throughput | 55 |
| 5.3 | CPU Usage | 56 |
| 5.4 | Round-trip Delay | 57 |
| 5.5 | Overhead | 58 |
| 5.5.1 | DPSLB Overhead | 58 |
| 5.5.2 | Load Balancing Process Overhead | 60 |
| 6 | Related Work | 62 |
| 6.1 | Dynamic Load Balancing in Distributed Content-based Publish/Subscribe | 62 |
| 6.2 | A DDS-compliant infrastructure for fault-tolerant and scalable data dissemination | 63 |
| 6.3 | Scalability of Distributed Dynamic Load Balancing Mechanisms | 64 |
| 6.4 | Discussion | 65 |
| 7 | Conclusion | 67 |
| 7.1 | Future Work | 68 |
| 8 | Bibliography | 70 |

List of Figures

| | |
|---|----|
| Figure 1 – Publication and Subscription Models [18] | 18 |
| Figure 2 – DDS System Architecture [16] | 18 |
| Figure 3 – Content Filtered Topic example [20] | 19 |
| Figure 4 – SDDL Architecture [26] | 21 |
| Figure 5 – Load balancing hierarchy [29] | 22 |
| Figure 6 – Taxonomy of dynamic load balancing algorithms [35] | 23 |
| Figure 7 – General properties of Autonomic Computing [47] | 25 |
| Figure 8 – MAPE-K control loop [49] | 26 |
| Figure 9 – DPS Load Balancing Architecture | 28 |
| Figure 10 – Example of data item objects | 31 |
| Figure 11 – A possible DPS flow and distribution | 32 |
| Figure 12 – DPS state transitions | 32 |
| Figure 13 – An example of Assignment Function applied upon data item | 33 |
| Figure 14 – Interactions between clients, Processing Nodes and Load Balancer | 35 |
| Figure 15 – Data flow before, during and after the Load Balancing Process | 36 |
| Figure 16 – Implementation architecture | 39 |
| Figure 17 – LoadBalancingAlgorithm interface | 42 |
| Figure 18 – Application listener interface | 45 |
| Figure 19 – Methods to create a Topic, Content Filtered Topic and Data Reader | 45 |
| Figure 20 – Slice transitions on a PN | 47 |
| Figure 21 – Interface Description Language (IDL) of the CacheTopic | 48 |
| Figure 22 – Generated CacheTopic Java Class | 49 |

| | |
|---|----|
| Figure 23 – Evaluation application topics | 51 |
| Figure 24 – Deployment of the evaluation application | 52 |
| Figure 25 – Deployment of the Virtual Machines | 53 |
| Figure 26 – Throughput over the time of the experiment (DI/s X seconds) | 55 |
| Figure 27 – CPU usage over the time of the experiment (% X seconds) | 56 |
| Figure 28 – Round-trip Delay over the time of the experiment (RTD in seconds X time in seconds) | 57 |
| Figure 29 – Mean throughput (DI/s) comparison among DPSLB solution and another without Load Balancing | 59 |
| Figure 30 – Mean Round-trip Delay comparison among DPSLB solution and another without Load Balancing | 59 |
| Figure 31 – Padres architecture [27] | 63 |
| Figure 32 – REVENGE high-level architecture [66] | 64 |
| Figure 33 – Multi-domain REVENGE distributed architecture [66] | 64 |

List of Tables

| | |
|---|----|
| Table 1 – Physical Machine specifications used in the experiment | 54 |
| Table 2 – Impact of the Load Balancing Process on RTD and throughput | 60 |
| Table 3 – Load Balancing Process overhead for different the numbers of <i>Slices</i> and data item production rates | 61 |