



Fortschritt- Berichte VDI

M.Sc. Volkan Gezer,
Kaiserslautern

NR. 874

A Modular and Scalable
Software Reference
Architecture for
Decentralized Real-Time
Execution on
Edge Computing

BAND
1|1

VOLUME
1|1

WS KL Werkzeugmaschinen
und Steuerungen
TU KAIERSLAUTERN

A Modular and Scalable Software Reference Architecture for Decentralized Real-Time Execution on Edge Computing

Dem Fachbereich Maschinenbau und Verfahrenstechnik
der Technischen Universität Kaiserslautern
zur Erlangung des akademischen Grades

Doktor-Ingenieur (Dr.-Ing.)

genehmigte
Dissertation

von
Herrn
M.Sc. Volkan Gezer
aus Istanbul

Tag der mündlichen Prüfung: 06.10.2021

Dekan: Prof. Dr.-Ing. Tilmann Beck

Promotionskommission:

Vorsitzende: Prof. Dr.-Ing. Kristin de Payrebrune
1. Berichterstatter: Prof. Dr.-Ing. Martin Ruskowski
2. Berichterstatter: Prof. Dr. Paul Lukowicz



REIHE 10
INFORMATIK/
KOMMUNIKATION

Fortschritt- Berichte VDI



M.Sc. Volkan Gezer,
Kaiserslautern

NR. 874

A Modular and Scalable
Software Reference
Architecture for
Decentralized Real-Time
Execution on
Edge Computing

BAND
1|1

VOLUME
1|1

WS **KL** Werkzeugmaschinen
und Steuerungen
TU KAIERSLAUTERN

Gezer, Volkan

A Modular and Scalable Software Reference Architecture for Decentralized Real-Time Execution on Edge Computing

Fortschritt-Berichte VDI, Reihe 10, Nr. 874. Düsseldorf: VDI Verlag 2021.

160 Seiten, 37 Bilder, 9 Tabellen.

ISBN 987-3-18-387410-1, ISSN 0178-9627,

57,00 EUR/VDI-Mitgliederpreis: 51,30 EUR

Für die Dokumentation: edge computing – echtzeit scheduling – fog computing – aufgabe übertragung – edge in produktion – edge server – software referenzarchitektur

Keywords: edge computing – real-time scheduling – fog computing – task offloading – edge in manufacturing –

edge server – software reference architecture

Edge Computing is expected to solve the latency and best-effort delivery problems of the renowned Cloud Computing. However, Edge Computing must be supported with a vendor-independent, scalable, and decentralized software reference architecture to fully exploit its benefits. This dissertation explains the enablers, requirements, and conceptual approach to create such architecture, and validates it with a framework to show its possibilities.

Bibliographische Information der Deutschen Bibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliographie; detaillierte bibliographische Daten sind im Internet unter www.dnb.de abrufbar.

Bibliographic information published by the Deutsche Bibliothek (German National Library)

The Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliographie (German National Bibliography); detailed bibliographic data is available via Internet at www.dnb.de.

D 386

Dissertation Technische Universität Kaiserslautern

Foreword

The basis for this thesis originally stemmed from my position as a Researcher at German Research Center for Artificial Intelligence (DFKI). Firstly, I would like to thank Professor Martin Ruskowski, for giving me the chance to work on the related projects that allowed me to perform research on this area and deepen my knowledge.

Additional thanks go to Prof. Dr.-Ing. Kristin de Payrebrune for chairing the examination board and to Prof. Dr. Paul Lukowicz for accompanying my examination as a referee. I also would like to thank my colleagues at DFKI - especially Andre Hennecke, Christiane Plociennik, Max Birtel, Moritz Ohmer, Pascal Rübel - for their valuable inputs and feedbacks to my work, also colleagues in Team Shannon for the working atmosphere, team collaboration, openness for discussion, and the team activities within or outside the working hours, and Dr. Achim Wagner for supporting me during structuring my thesis with status meetings, and corrections.

Special thanks go to my family: my mother Billur, my father Sebahattin, and my wife, Elif Nur, who kept me motivated and patiently supported me during the stressful times.

Volkan Gezer

Contents

Foreword	III
Abstract	VII
Kurzfassung	VIII
List of Abbreviations	IX
1 Introduction	1
1.1 Problem Definition	5
1.2 Objectives	8
1.3 Approach	10
2 State of the Art	11
2.1 Cloud Computing and Edge Computing	11
2.1.1 Related Work	17
2.1.2 Requirements	23
2.1.3 Enablers	25
2.2 Real-Time Computing	32
2.2.1 Challenges in Real-Time Systems	34
2.2.2 Scheduling for Real-Time Processing	39
2.3 Summary	54
3 Scheduling and Decision Making Methodology	55
3.1 NAPATA Scheduling	55
3.2 Problem Formulation on Server Selection	61
3.3 Summary	64
4 Software Reference Architecture	65
4.1 Edge Server and End Device Concepts	68
4.2 Service and Task	72
4.2.1 Legacy	74
4.2.2 Simple	74
4.2.3 Simple Periodic	74

4.2.4	Service Behaviours	75
4.2.5	Service Parameters	76
4.3	Decision Making	78
4.4	Summary	82
5	Implementation and Validation	83
5.1	Edge Server Components	83
5.1.1	Configurator	85
5.1.2	TCP Server	88
5.1.3	Message Router	88
5.1.4	Security Protocols	89
5.1.5	Resource Monitor	89
5.1.6	Orchestrator	90
5.1.7	Virtual Processors	92
5.1.8	Other Components	93
5.2	Communication	93
5.3	Standard Commands	96
5.4	Requesting Tasks	98
5.5	Topology Designer	103
5.6	Edge Server Creation	105
5.7	Validation of Framework	107
5.8	Summary	115
6	Conclusion and Outlook	116
6.1	Conclusion	116
6.2	Outlook	118
A	Appendix	120
A.1	Edge Topology Designer File Example	120
A.2	Creation of an Edge Server Using RTEF: An Example	122
A.3	Definitions	125
Bibliography		129
Lebenslauf		

Abstract

Cloud Computing, or the Cloud, became one of the most used technologies in today's world, right after its possibilities had been figured out. It is a renowned technology that enables ubiquitous access to tasks that need collaboration or remote monitoring. It is widely used in daily lives as well as the industry. The paradigm uses Internet Technologies which rely on best-effort communication. Best-effort communication limits the applicability of the technology in the domains where the timing is critical. Edge Computing is a paradigm that is seen as a complementary technology to the Cloud. It is expected to solve the Quality of Service (QoS) and latency problems that are raised due to the increased count of connected devices, and the physical distance between the infrastructure and devices. The Edge Computing adds a new tier between Information Technology (IT) and Operational Technology (OT) and brings the computing power close to the source of the data. Computing power near devices reduces the dependency to the Internet; hence, in case of a network failure, the computation can still continue. Close proximity deployments also enable the application of Edge Computing in the areas where real-timeliness is necessary. Computation and communication in Edge Computing are performed via Edge Servers. This thesis suggests a standardized and hardware-independent software reference architecture for Edge Servers that can be realized as a framework on servers, to be used on domains where the timing is critical. The suggested architecture is scalable, extensible, modular, multi-user supported, and decentralized. In decentralized systems, several precautions must be taken into consideration, such as latencies, delays, and available resources of the neighbouring servers. The resulting architecture evaluates these factors and enables real-time execution. It also hides the complexity of low-level communication and automates the collaboration between Edge Servers to enable seamless offloading in case of a need due to lack of resources. The thesis also validates an exemplary instance of the architecture with a framework, called Real-Time Execution Framework (RTEF), with multiple scenarios. The tasks used are resource-demanding and requested to be executed on an Edge Server in an Edge Network comprising multiple Edge Servers. The servers can make decisions by evaluating their availabilities, and determine the optimal location to execute the task, without causing deadline misses. Even under a heavy load, the decisions made by the servers to execute the tasks on time were correct, and the concept is proven.

Kurzfassung

Cloud Computing, oder die Cloud, wurde zu einer der meistgenutzten Technologien in der heutigen Welt, gleich nachdem ihre Möglichkeiten entdeckt wurden. Es handelt sich um eine anerkannte Technologie, die einen ubiquitären Zugriff auf Aufgaben ermöglicht, die Zusammenarbeit oder Fernüberwachung erfordern. Sie ist sowohl im täglichen Leben als auch in der Industrie weit verbreitet. Das Paradigma nutzt Internet-Technologien, die auf Best-Effort-Kommunikation beruhen. Die Best-Effort-Kommunikation schränkt die Anwendbarkeit der Technologie in den Bereichen ein, in denen das Timing kritisch ist. Edge Computing ist ein Paradigma, das als eine ergänzende Technologie zur Cloud gesehen wird. Probleme mit der Dienstgüte (QoS) und Latenzen, die durch die steigende Anzahl angeschlossener Geräte und der physischen Entfernung zwischen Infrastruktur und Geräten entstehen, sollen dadurch gelöst werden. Das Edge Computing fügt eine neue Ebene zwischen der Informationstechnologie (IT) und der Betriebstechnologie (OT) hinzu und bringt die Rechenleistung nahe an die Quelle der Daten. Das Näherbringen der Geräte reduziert die Abhängigkeit vom Internet und kann somit Berechnung auch bei einem Netzausfall sicherstellen. Ebenso kann dadurch das Einsatzgebiet des Edge Computing um Bereiche erweitern, in denen Echtzeitfähigkeit gefordert ist. Berechnung und Kommunikation im Edge Computing werden über Edge Server durchgeführt. Diese Dissertation schlägt eine standardisierte und hardwareunabhängige Software-Referenzarchitektur für Edge Server vor, die als Framework auf Servern realisiert werden kann, um sie in zeitkritischen Domänen einzusetzen. Die vorgeschlagene Architektur ist skalierbar, erweiterbar, modular, mehrbenutzerfähig und dezentralisiert. In dezentralen Systemen müssen verschiedene Maßnahmen berücksichtigt werden, wie z.B. Latenzen, Verzögerungen und verfügbare Ressourcen der benachbarten Server. Die resultierende Architektur wertet diese Faktoren aus und ermöglicht die Ausführung in Echtzeit. Sie kapselt auch die Komplexität der Low-Level-Kommunikation und automatisiert die Zusammenarbeit zwischen Edge-Servern, um ein reibungsloses Offloading zu ermöglichen, falls ein Bedarf aufgrund von Ressourcenmangel besteht. Die Dissertation validiert auch eine exemplarische Instanz der Architektur mit einem Framework, genannt Real-Time Execution Framework (RTEF), mit mehreren Szenarien. Die verwendeten Aufgaben sind ressourcenintensiv und sollen auf einem Edge-Server in einem Edge-Netzwerk mit mehreren Edge-Servern ausgeführt werden. Die Server können durch Auswertung ihrer Verfügbarkeiten Maßnahmen ergreifen und den optimalen Ort für die Ausführung der Aufgabe bestimmen, ohne dass es zu Terminüberschreitungen kommt.

Abbreviations

API	Application Programming Interface
CDN	Content Delivery Networks
CM	Cloud Manufacturing
CPS	Cyber-Physical Systems
CPU	Central Processing Unit
EDF	Earliest Deadline First
ETD	Edge Topology Designer
GPOS	General-Purpose Operating System
HPC	High-Performance Computing
HTTP	HyperText Transfer Protocol
I/O	Input/Output
IoT	Internet of Things
ISO	International Organization for Standardization
IT	Information Technology
LAN	Local Area Network
LST	Least Slack Time
NAPATA	Non-resumable And Preemptible Aperiodic TAsk
OS	Operating System
OT	Operational Technology
POSIX	Portable Operating System Interface
PSC	program/software/command
QoS	Quality of Service
RTEF	Real-Time Edge Framework
RTOS	Real-Time Operating System
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
UDP	User Datagram Protocol
VM	Virtual Machine
VP	Virtual Processor
WCET	Worst-Case Execution Time
WCEU	Worst-Case Execution Utilization

