ABSTRACT

The emerging edge computing paradigm promises to deliver superior user experience and enable a wide range of Internet of Things (IoT) applications. In this paper, we propose a new market-based framework for efficiently allocating resources of heterogeneous capacity-limited edge nodes (EN) to multiple competing services at the network edge. By properly pricing the geographically distributed ENs, the proposed framework generates a market equilibrium (ME) solution that not only maximizes the edge computing resource utilization but also allocates optimal resource bundles to the services given their budget constraints. When the utility of a service is defined as the maximum revenue that the service can achieve from its resource allotment, the equilibrium can be computed centrally by solving the Eisenberg-Gale (EG) convex program. We further show that the equilibrium allocation is Pareto-optimal and satisfies desired fairness properties including sharing incentive, proportionality, and envy-freeness. Also, two distributed algorithms, which efficiently converge to an ME, are introduced. When each service aims to maximize its net profit (i.e., revenue minus cost) instead of the revenue, we derive a novel convex optimization problem and rigorously prove that its solution is exactly an ME. Extensive numerical results are presented to validate the effectiveness of the proposed techniques.

TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **Sl. No** | **Title** | **Page No.** |
|  | **Abstract** | **i** |
|  | **Table of Contents** | **ii** |
|  | **List of Figures** | **iv** |
|  | **List of Tables** | **v** |
|  | **Abbreviation** | **vi** |
| **1.** | **INTRODUCTION** | **1-3** |
|  | 1.1 Introduction | 1 |
|  | 1.2 Existing System | 2 |
|  | 1.3 Proposed System | 2 |
|  | 1.4 Advantages of proposed system | 3 |
| **2.** | **REQUIREMENT ANALYSIS** | **4** |
|  | 2.1 Requirement Analysis | 4 |
|  | 2.2 Requirement Specification  2.2.1 Functional Requirement | 4  4 |
|  | 2.3 Computational Resource requirements  2.3.1 Hardware requirements  2.3.2 Software requirements | 4  4  4 |
| **3** | **DESIGN** | **5-9** |
|  | 3.1 Introduction | 5 |
|  | 3.2 Use case Diagram | 6 |
|  | 3.3 Class Diagram | 7 |
|  | 3.4 Sequence Diagram | 8 |
|  | 3.5 State Chart Diagram | 9 |
| **4** | **MODULES** | **10-11** |
|  | 4.1 Modules | 10 |
| **5** | **IMPLEMENTATION** | **12** |
|  | 5.1 Sample Code | 12 |
|  | 5.2 Sample Database  5.2.1 Auth Group  5.2.2 Auth Permission  5.2.3 Consumer Order | 13  13  13  14 |
| **6** | **SCREENSHOTS** | **15-18** |
|  | 6.1 Screenshots  6.1.1 Login Page  6.1.2 Registration Page  6.1.3 Home Page  6.1.4 Add Product Page | 15  15  16  17  18 |
| **7** | **TESTING** | **19-21** |
|  | 7.1 Overview of Testing | 19 |
|  | 7.2 Typing of Testing  7.2.1 Unit Testing  7.2.2 Integration Testing  7.2.3 Functional Testing | 19  19  19  20 |
|  | 7.3 Unit Testing | 20 |
|  | 7.4 Integration Testing | 21 |
|  | 7.5 Acceptance Testing | 21 |
| **8** | **CONCLUSION AND FUTURE WORK** | **22-23** |
|  | 8.1 Conclusion | 22 |
|  | 8.2 Scope for future work | 22 |
| **9** | **REFERENCES** | 24 |

LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **Sl No** | **Title** | **Page No.** |
| 3.1 | Architecture | 8 |
| 3.2 | Use case Diagram | 6 |
| 3.3 | Class Diagram | 7 |
| 3.4 | Sequence Diagram | 8 |
| 3.5 | State chart Diagram | 9 |
| 6.1 | Login Page | 15 |
| 6.2 | Registration Page | 16 |
| 6.3 | Home Page | 17 |
| 6.4 | Add Product Page | 18 |

LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| **Sl No** | **Title** | **Page No.** |
| 5.1 | Auth Group | 13 |
| 5.2 | Auth Permission | 13 |
| 5.3 | Customer Order | 14 |

ABBREVIATION

|  |  |
| --- | --- |
| GUI | GRANPHICAL USER INTERFACE |
| PY | PYTHON |
| OPP | OBJECT ORIENTED PROGRAMMING |
| SQL | STRUTURAL QUERY LANGUAGE |
| DB | DATABASES |
| HTML | HUPER TEXT MARKUP LANGUAGE |
| CSS | CASCATING STYLE SHEET |