

アプリケーションと連携する MEC プラットフォームの実装評価

澤田 知貴[†] 藤井 雅章^{††} 丸山 亮^{††} 山田 智紀^{††} 藤井 宏行^{††}

福田 利道^{††} 橘 拓至[†]

[†] 福井大学 大学院工学研究科 〒910-8507 福井県福井市文京 3-9-1

^{††} 富士通株式会社 〒211-8588 神奈川県川崎市中原区上小田中 4-1-1

E-mail: [†]kazuki-s@network.fuis.u-fukui.ac.jp,

^{††}{fujii_masaaki,maruyama.ryo,yamada.tomon-01,fujii.hiroyuki,t_fukuda}@fujitsu.com, ^{†††}takuji-t@u-fukui.ac.jp

あらまし MEC プラットフォーム環境では MEC サーバとクラウドサーバを併用し、各タスクを適切に各サーバへ割り当ててことでアプリケーションの性能を維持できる。しかしながら、多数の非リアルタイムタスクを MEC サーバで処理することは難しく、MEC プラットフォーム環境では、リアルタイムタスクと非リアルタイムタスクを状況に応じて動的にサーバへ割り当てる必要がある。本稿では、MEC 環境において MEC サーバとクラウドサーバを併用する多人数情報共有型アプリに対して、MEC 環境の状況に応じて両サーバを柔軟に使用する動的タスク割当アルゴリズムを実装する。提案技術では、本アプリで処理される全タスクを、リアルタイム性を要求するタスクとリアルタイム性を要求しないタスクに分類し、さらにリアルタイム性の要求するタスクを高優先タスクと低優先タスクに分類する。そして、提案技術では、MEC プラットフォームの伝送遅延・処理遅延の情報をアルゴリズムで用い、低優先タスクの割当先を決定する。本技術を実装したアプリとサーバを用いて実機実験を行い、動的タスク割当技術の性能を評価する。

キーワード MEC, クラウド, サーバ, アプリケーション, タスク割当

Evaluation of the Implementation of MEC Platform Cooperated with Applications

Kazuki SAWADA[†], Masaaki FUJII^{††}, Ryo MARUYAMA^{††}, Tomonori YAMADA^{††}, Hiroyuki FUJII^{††},
Toshimichi FUKUDA^{††}, and Takuji TACHIBANA[†]

[†] Graduate School of Engineering, University of Fukui 3-9-1 Bunkyo, Fukui, Fukui 910-8507, Japan

^{††} Fujitsu limited 4-1-1 Kamikodanaka, Nakahara, Kawasaki, Kanagawa 211-8588, Japan

E-mail: [†]kazuki-s@network.fuis.u-fukui.ac.jp,

^{††}{fujii_masaaki,maruyama.ryo,yamada.tomon-01,fujii.hiroyuki,t_fukuda}@fujitsu.com, ^{†††}takuji-t@u-fukui.ac.jp

Abstract In the multi-access edge computing (MEC) platform environment, the MEC server and the cloud server are used together, and the performance of the application can be maintained by appropriately allocating each task to each server. However, it is difficult for the MEC server to process a large number of non-real time tasks due to its low processing performance. In an MEC platform environment, it is necessary to dynamically allocate real time and non-real time tasks to the server according to the situation. In this paper, we implement a dynamic task assignment algorithm that flexibly uses both MEC and cloud servers depending on the situation of the MEC environment for a multi-user information-sharing application. In the proposed method, all tasks processed by this application are classified into tasks that require real-time performance and tasks that do not require real-time performance, and real-time tasks are classified into high-priority tasks and low-priority tasks. In the proposed method, the transmission and processing delay information of the MEC platform is used in the algorithm to determine where to allocate low-priority tasks. We evaluate the performance of the dynamic task assignment algorithm by conducting experiments with an application and a server implementing the task assignment.

Key words MEC, Cloud, Server, Application, Task Assignment