

発表概要

TAKAO：高速不揮発メモリ構成下における 永続データのための Scala ライブラリ

明島 利樹¹ 山崎 憲一¹

2015年11月6日発表

近年、高速不揮発性メモリ（NVM）の性能が向上しており、将来的には従来の DRAM と同等のアクセス速度になると考えられている。本発表では、既存のハードディスクと DRAM がすべて NVM に置き換えられることを想定する。その場合、メモリ上のデータはそのまま保持されるが、それだけではデータの永続化は実現できない。本発表で提案するライブラリ TAKAO は、NVM 上での永続データのためのライブラリであり、2つの機能を持つ。1つはメモリ上のオブジェクトにグローバルな識別子を付与することで、そのデータを次の実行時に再利用可能にする機能である。もう1つは、電源喪失などの障害時でもデータの一貫性を保つ機能である。TAKAO は、関数型言語 Scala 上で提供される。関数型言語のもつ参照透過性のため、基本的にデータが書き換えられなくなり、データの一貫性を保つことが容易となる。また、副作用をともなう操作に対しては、原子的更新の機能を導入する。Scala は Java 仮想マシン（JVM）上で動作するが、JVM を NVM 上に実装するには次の課題がある。1つは、永続化データの選別である。これには永続化データとその内部表現に必要なデータだけを残し、他を回収するための Recovery GC を実装した。もう1つは、原子的更新の実装である。これにはソフトウェアトランザクショナルメモリと類似の技術を用いた。本発表では提案するライブラリ TAKAO の設計と詳細な実装について述べたのち、性能を評価する。

TAKAO: A Scala Library for Persistent Data on Non-volatile Memory Based Systems

TOSHIKI AKEHATA¹ KENICHI YAMAZAKI¹

Presented: November 6, 2015

The performance of high-speed NVM (Non-Volatile Memory) is improved, and its access speed will be the same as DRAM in the near future. In this paper, we assume a computer architecture in which the main memory and the hard disk are replaced by the NVM. Although the NVM inherently keeps the written data ever, it is not enough for data persistence. TAKAO, proposed in this paper, is a library for persistent data on NVM. It provides a global naming function to a persistent data, using which the data can be accessed in the subsequent execution. TAKAO also provides a data protection mechanism to keep data consistency even when a power failure occurs. TAKAO is provided on top of a functional programming language Scala. Since Scala has a referential transparency which means that data is not updated destructively, it is easy to keep data consistency for persistent data. We introduce an atomic update mechanism for data modification as well. Scala runs on JVM (Java Virtual Machine), and the JVM has technical issues to be implemented on NVM. One is how to separate persistent data with others. We propose a recovery GC which marks necessary internal data for persistent data and collects useless data. As for another issue, an atomic update implementation, we use a similar technique as software transactional memory. In this paper, we describe the design and detailed implementation of TAKAO, and finally show the performance evaluation.

¹ 芝浦工業大学大学院理工学研究科
Shibaura Institute of Technology, Graduate School of Engineering and Science, Minato, Tokyo 108-8548, Japan