

発表概要

Scala Actor ライブラリによる Safe Ambients フレームワークの実装

岡田 翔太^{1,a)} 馬谷 誠二¹ 林 奉行¹ 八杉 昌宏¹ 湯浅 太一¹

2012年1月23日発表

アンビエント計算はプロセス代数の一種であり、並行プロセス間の協調動作や計算機間のコード移動は、アンビエントの移動動作として表現される。アンビエントの階層を用いることで、LAN、PC クラスタ、マルチコアプロセッサ、モバイルエージェントなどを統一的に表現可能な点の特徴である。本発表では、アンビエント計算の一種である Safe アンビエントに基づく、並列・分散計算のための Scala フレームワークを提案する。本フレームワークを用いると、分散環境の構成要素をアンビエントとして統一的に記述し、アンビエントの形で実行できる。アンビエントは通常のオブジェクトとして記述されるため、各アンビエントにインスタンス変数やメソッドを持たせることができ、分かりやすく簡潔なプログラムを記述することができる。Scala を用いることで、従来の Java によるフレームワークに比べ、アンビエントの定義をより簡潔に記述できるようになっている。異なる計算機間のアンビエントの移動は、Scala のクロージャオブジェクトを受け渡すことにより実現している。Safe アンビエントでは、アンビエント全体を1つの Scala アクタとして実行するのが望ましい場合と、それに含まれるプロセスを別々のアクタとして実行するのが望ましい場合がある。本フレームワークでは、アンビエントの型を用いて、どちらの方法により実行するかを適切に選択する。

Implementation of the Safe Ambients Framework Using Scala Actor Library

SHOTA OKADA^{1,a)} SELJI UMATANI¹ TOMOYUKI HAYASHI¹
MASAHIRO YASUGI¹ TAIICHI YUASA¹

Presented: January 23, 2012

The Ambient calculus is a kind of process calculi. Code migration among computers and synchronized movement among parallel processes are represented as movements of ambients. LANs, PC clusters, multicore processors and mobile agents are uniformly represented as components of the hierarchy of ambients. In our presentation, we propose a Scala framework for parallel and distributed computing, based on Safe Ambients. Each component of distributed environments is uniformly represented and executed as an ambient. Since each ambient is a normal Scala object, it can contain instance variables and instance methods so that we can write programs in a practical manner. By implementing the framework in Scala, we can write definitions of ambients easier than the previous framework implemented in Java. Movements of ambients among different computers are achieved by delivering Scala's closure objects. From the study of Safe Ambients, we find that there are two ways for executing ambients using Scala actors; each ambient should be executed using only a single actor, or multiple processes included in one ambient should be executed using different actors. Our framework chooses the suitable way for executing each ambient using its type information.

¹ 京都大学大学院情報学研究科
Graduate School of Informatics, Kyoto University, Kyoto
606-8501, Japan

^{a)} okadas@kuis.kyoto-u.ac.jp