Title	ネットワークシミュレータSSFNetの分散プログラミングフレームワークNekoへの統合
Author(s)	松下，誠和
Citation	
Issue Date	2005-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1919
Rights	
Description	Supervisor:Defago Xavier，情報科学研究科，修士

Integrating The Neko Distributed Programing Framework with The SSFNet Network Simulator

Tomokazu Matsushita (310105)

School of Information Science,
Japan Advanced Institute of Science and Technology

February 10, 2005

Keywords: distributed protocol, java, network simulator, atomic broadcast, distributed consensus, performance evaluation.

Performance engineering is an important activity in construction of complex systems, including distributed systems. Two common approaches to evaluating the performance of distributed algorithms is simulation runs in a simulated execution environment, and measurements in a real environment, and solid studies with reliable results often include both. A major factor that contributes to this complexity is the fact that there are few environments to support both approaches.

The Neko framework, developed at the Swiss Federal Institute of Technology (EPFL) and the Japan Advanced Institute of Science and Technology (JAIST) answers this problem. It is a Java distributed programming framework to develop distributed algorithms and to evaluate their performance. Developers can develop and evaluate distributed algorithms both in a real environment and a simulated environment, using the same source code. Neko also offers a simple message passing interface and a hierarchy of layers to ease the construction of algorithms.

The current version of Neko, however, has an important shortcoming: it lacks simulated networks that are based on realistic models of networks, and thus realistic simulations are not possible. In order to solve this problem, I integrated Neko with the SSFNet, a widely used network simulator with realistic network models that focuses on scalable simulations of large systems. Concretely, integration is performed by offering a Neko interface to SSFNet's simulated networks. Problems to solve during the integration included the following: (1) synchronizing the scheduler of Neko and that of SSFNet; (2) mapping between two different addressing schemes, used by each framework; and (3) converting messages at the interface between the frameworks. My code has been accepted into the Neko codebase, and will be part of future releases of Neko.

I evaluated the performance of two consensus algorithms on atomic broadcast with the integrated Neko framework, repeating an earlier study that used a different simulation model. Atomic broadcast and consensus are communication primitives useful in the construction of fault-tolerant distributed systems. One algorithm uses a centralized and the other a decentralized communication pattern. I used an SSFNet simulation model that describes a simple switched Ethernet network, and could successfully reuse the algorithms implemented in Neko, thus validating my integration work. The results in the original study have been obtained with a simple abstract model with a parameter that had three different settings. My results, obtained with a realistic model, match one of these sets of results well, and I could decide with their help which parameter setting in the original study yielded the model closest to a switched Ethernet network.