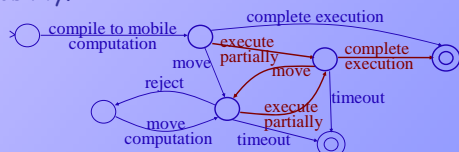


Extensions to and Optimization of a Mobile Computation Language System for Internet Programming

S. Monisha Pulimood, Gregory Adkins, Elizabeth Carter, Amanda Micai
The College of New Jersey

I. Background

- Mobile Computations facilitate sharing of resources on the Internet.
- Programs that can commence execution at one site, halt execution, migrate and continue execution at another site without needing to start over.
- Must carry its closure (state of execution) - strong mobility.



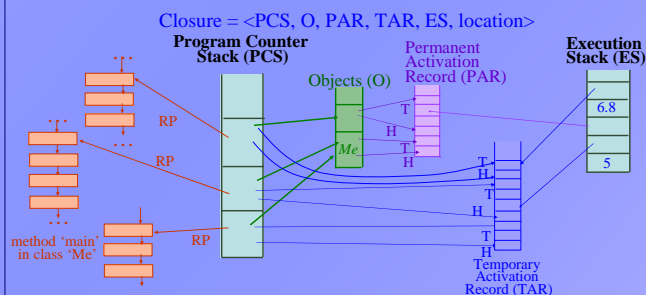
- Native support for strongly mobile computations is essential for efficiency and security.
- Provided by the Mobile Computational Model that includes the Mobile Computational Language (MCL), for developing mobile computation applications, and the RISC-based Virtual Machine (RVM).

II. Objectives

- Apply optimization techniques to improve efficiency
- Add functionality to support data structures like multi-dimensional arrays
- Design and implement a communication model

III. Optimization

- Compiler level optimization
 - Programs with deeply nested loop structures optimized in order to eliminate slow downs in execution.
 - Dead code removed in order to reduce space usage and transmission overheads.
- Optimization of RVM implementation
 - Indexed linked lists in place of standard linked lists
 - Change the representation of values to avoid myriad conversions.

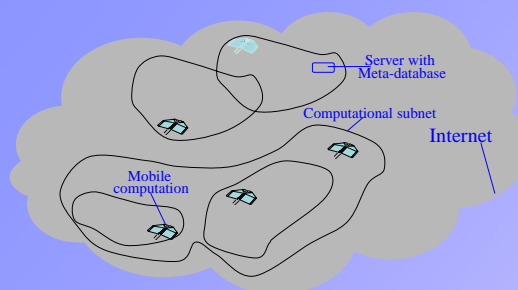


IV. Extensions to the Language

- The syntactical structure for array declaration and manipulation is similar to that of other popular programming languages.
- Must implement efficient usage of borrowed resources.
- Illegal memory access and run-time errors should be avoided.
- Indirection is the preferred method of storing arrays.

V. The Communication Model

- Existing mobile agent toolkits require trade-offs.
 - Most are Java-based, therefore closure capture is not supported.
 - Some Java-based toolkits use modified versions of JVM.
 - Non-Java-based toolkits extend existing systems.
- Our Model:
 - Specifically designed to support mobile computations with strong mobility.
 - The hosts that a mobile computation can execute on form a 'computational subnet'.



- Communication between computations is through message passing
- Our model implements the asynchronous protocol with additional checks to assure reliability, and sockets instead of RMI.
- The model implements the push migration strategy.

VI. Conclusions

- The tasks undertaken in this project are non-trivial
- Complicated by considerations of strong mobility.
- Constraints due to the need for increased security and reliability in the presence of heterogeneous architectures and networks.
- The next phase involves detailed testing and analysis of the effect of the extensions and optimization efforts.