Rapid prototyping of Matlab/Java Distributed Applications using the *JavaPorts* components framework

Elias S. Manolakos

Electrical and Computer Engineering Department
Northeastern University, 442 Dana Research Building, Boston MA 02115
Tel: 617-3733021, Fax: 617-3738970, E-mail: elias@ece.neu.edu

Country of Citizenship: USA
Areas:
Networked Embedded Systems
Software Architecture, Reusablity, Scalability, and Standards
Middleware Libraries and Application Programming Interfaces
Preference: Oral Presentation

Overview

Although PC clusters have become a commodity, there is still a lack of software technologies for the rapid prototyping and evaluation of end-to-end distributed computation strategies for heterogeneous clusters. Such applications may involve a large number of tasks (depicted as small circles in the task graph of Figure 1) which interact while running in the different nodes of a heterogeneous network, that possibly includes smart sensors, embedded controllers, various types of PCs, compute/data servers etc.

At Northeastern University we are developing *JavaPorts* (JP) [1] [2], a novel component-based distributed processing framework which provides to application developers:

- The capability to create, modify and reuse across applications and platforms software components for implementing communicating tasks.
- Abstractions, Application Programming Interfaces (API) and associated middleware for inter-task *anonymous* message passing. Two tasks may exchange information without naming each other explicitly, or having to know each other's location in the network.
- Tools for the easy building of distributed applications as component assemblies and for launching them in a network.

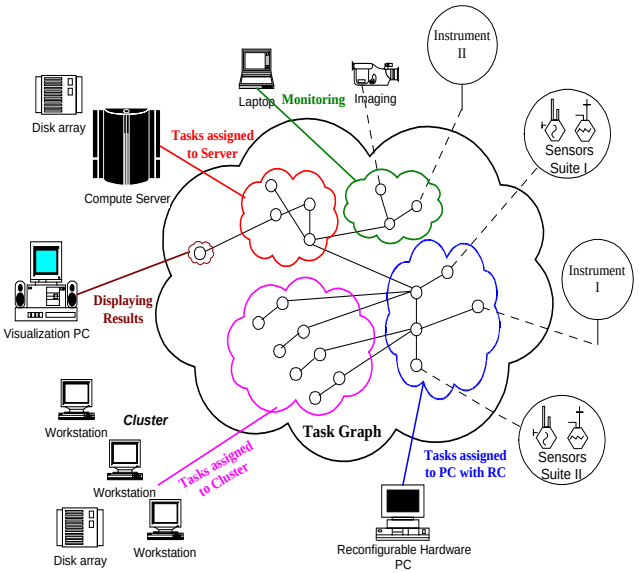

Figure 1. A distributed network application consists of several concurrent tasks (circles) that interact (lines) and form a complex *task graph*. Tasks are allocated to heterogeneous resources in the network.

JavaPorts makes it easy for a team of collaborators with limited expertise in distributed processing to: (a) define an application in a top-down fashion, (b) to generate software components (for each task), (c) to execute the components at any node in the network without having to change any part of the source code. The high level of *task location transparency* supported by JP minimizes the human effort and the cost of software re-engineering, thus allowing more application design iterations in a fixed amount of time. During the last year we have completed a new JP software release v 2.1 in which we added:

- a graphical tool for visual application (task graph) development and management.
- an automatically implemented scheme for launching and terminating gracefully a distributed application. Releasing the code developer from the distributed termination responsibility removes a major headache in developing programs for clusters (as it can be attested by users of message passing systems such as MPI and PVM).
- Run-time support for a variety of networks of workstations, including Linux- and Solaris-based clusters (with and without NFS capabilities)

With Matlab v. 6 now available (which supports Java) we have also achieved the full integration of Matlab components into the *JavaPorts* framework. A developer can mix, into the same distributed application, any number of Matlab and Java components that may interact directly (not via files) and pass messages *anonymously* to each other using the *JavaPorts* communication API. Such a capability is essential because although the Matlab code base is increasing rapidly there is still no easy mechanism for integrating Matlab functions into a distributed computation without investing a substantial effort in re-coding. *JavaPorts* not only allows us to implement arbitrary patterns of component interactions but also supports the development of mixed Java/Matlab distributed applications for heterogeneous clusters with computers running under different operating systems and possibly including reconfigurable hardware boards (FPGAs) to accelerate a specific part of the overall network computation (joint work with Prof. M. Leeser, Northeastern University [3]). In the presentation we will discuss different application examples and present encouraging performance data.

References

[1] E. S. Manolakos, D. Galatopoullos, A. Funk, "Component-based peer-to-peer distributed processing in heterogeneous networks using *JavaPorts*", Proceedings of the IEEE Conference on Network Computing, pp. 234-237, February 2002.

[2] E. S. Manolakos, D. Galatopoullos, A. Funk, "*JavaPorts:* An Environment for the Rapid Prototyping of Heterogeneous Network-centric Distributed Processing Applications" in the Proceedings of the Fifth Workshop on High Performance Embedded Computing (HPEC-01), pp. 149-151, MIT Lincoln Labs, Lexington, MA, November 2001.

[3] L. A. King, H. Quinn, M. Leeser, D. Galatopoullos, E.S. Manolakos, "Runtime Execution of Reconfigurable Hardware in a Java Environment" in the Proceedings of the IEEE International Conference on Computer Design (ICCD-01), pp. 380-385, September 2001.