Chapter 18

PERFORMANCE OF NETWORK-BASED PROBLEM-SOLVING ENVIRONMENTS

Rajini I. Balay, Mladen A. Vouk and Harry Perros *

Abstract The Network-Based Problem Solving Environment (NBPSE) paradigm is emerging as an important approach to complex problem solving, management and decision making. The quality of service and the performance of an NBPSE depend heavily on the communication infrastructure and middleware used to construct it. In this paper we discuss communication related performance issues through examples drawn from an operational NBPSE, a commercial middleware product, and a custom-made middleware designed for development of NBPSE prototypes. For example, in an IP-based high performance network-oriented environment, functionally extensive commercial communication middleware may operate as much as 2 or more times slower than specialized lightweight middleware. Furthermore, the way the problem solution is distributed and accessed, as well as the user activity level, can significantly impact field performance and the scalability of an NBPSE.

1. INTRODUCTION

A Problem Solving Environment (PSE) is a computer-based system that provides all the computational facilities necessary to solve a target class of problems efficiently. Modern network-based PSEs (NBPSE) are collections of distributed, cooperating interfaces, libraries, databases, programs, tools, clients, and intelligent agents which facilitate user interaction and cooperative execution of the components charged with the solution tasks (Gallopoulos et al., 1994; Rice and Boisvert, 1996; Singh and Vouk, 1999).

The need for effective and efficient communication among the PSE components (or objects) is obvious. This has resulted in a proliferation of commu-

*This work was supported in part by MCNC/EPA grant C94-7050-027/029, NSF award ACS-9696131, IBM SUR grants and CACC award 5-3007
nunciation building blocks, or middleware, for distributed scientific and other computing and problem solving. For effective NBPSE operation, a complete version of such middleware needs to support actively at least the following:

1) unicast and multicast messaging, 2) synchronous and asynchronous message processing, 3) efficient transfer of high-volume data, 4) ability to import information from and export information to processes that are not part of the NBPSE, 5) ability to invoke functions of a subsystem from any other process, and 6) last, but not least, end-to-end quality of service (QoS) (Balay, 1998). In many cases communication is based on the TCP/UDP/IP stack. Two examples of IP-based middleware are PVM and MPI. A large group of systems are also based on the CORBA-compliant commercial object brokers (Singh and Vouk, 1999). While middleware can greatly simplify development of NBPSEs, it usually introduces extra overhead. This overhead can be accentuated in environments where the resources (such as workstations and supercomputers) are interconnected by high-speed links (e.g., switched 100, 155 or 1000 Mbps links) because any middleware communication deficiencies may become a major performance bottleneck.

In general, NBPSE communication performance is a direct function of a) its communication and data management middleware, b) its architecture and topology, c) its application-level algorithms and solutions, and d) its usage patterns (including the number and distribution of users). In this paper we illustrate some issues related to items a), b) and d) using a combination of simulation and measurement-based examples from a real operational NBPSE called EDSS (based on special-purpose middleware called LSB), and from a CORBA based middleware suite called SOMobjects. In Section 2 we describe our empirical environment. In Section 3, we discuss our experiments, simulations and results. Conclusions are in Section 4.

2. **EMPIRICAL ENVIRONMENT**

The **Environmental Decision Support System** (EDSS) is an operational NBPSE developed by the MCNC Environmental Programs Group in cooperation with the US EPA and NC State University (Ambrosiano et al., 1995). EDSS was developed to help research scientists and environmental planners and managers model and evaluate environmental quality issues and make decisions at different levels of granularity. It was used as a prototype for the EPA's Models-3 (Dennis et al., 1996) system. EDSS runs on Unix and PC platforms and it is in daily use at a number of sites in the U.S.A. and world-wide.

EDSS supports air quality modeling through three principal subsystems, a data and model engineering and configuration management subsystem, a simulation planning and execution management subsystem, and an analysis and visualization subsystem. The simulation planning subsystem, called the