IDEA GROUP PUBLISHING
701 E. Chocolate Avenue, Hershey PA 17033-1117, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com
ITB7302

Chapter I

Managing Real-Time Distributed Multimedia Applications

Vana Kalogeraki
Hewlett-Packard Laboratories, USA

Peter Michael Melliar-Smith and Louise E. Moser
UC Santa Barbara, USA

Distributed multimedia applications are characterized by timing constraints and end-to-end quality of service (QoS) requirements, and therefore need efficient management mechanisms to respond to transient changes in the load or the availability of the resources. This chapter presents a real-time distributed multimedia framework, based on the Common Object Request Broker Architecture (CORBA), that provides resource management and Quality of Service for CORBA applications. The framework consists of multimedia components and resource management components. The multimedia components produce multimedia streams, and combine multimedia streams generated by individual sources into a single stream to be received by the users. The resource management components provide QoS guarantees during multimedia transmissions based on information obtained from monitoring the usage of the system's resources.

INTRODUCTION

Real-time distributed multimedia environments have set forth new challenges in the management of processor and network resources. High-speed networks and powerful end-systems have enabled the integration of new types of multimedia applications, such as video-on-demand, teleconferencing, distance learning and collaborative services, into today's computer environments. Multimedia applications are variable in nature, as they handle a combination of continuous data (such as audio and video) and discrete data (such as text, images and control information) and impose strong requirements on data transmission, including fast transfer and substantial throughput.

This chapter appears in the book, Multimedia Networking: Technology, Management and Applications by Syed Mahbubur Rahman.

The technical requirements necessary to achieve timeliness are obviously more difficult to satisfy in distributed systems, mostly because of the uncertain delays in the underlying communication subsystem. This difficulty is further exacerbated by the heterogeneity of today's systems with respect to computing, storage and communication resources and the high levels of resource sharing that exist in distributed systems. Multimedia tasks may involve components located on several processors with limited processing and memory resources and with shared communication resources. Different transport mechanisms, such as TCP or UDP, can be used for data transfer within local- or wide-area networks.

Distributed object computing (DOC) middleware is software built as an independent layer between the applications and the underlying operating system to enable the applications to communicate across heterogeneous platforms. At the heart of the middleware resides an object broker, such as the OMG's Common Object Request Broker Architecture (CORBA), Microsoft's Distributed Component Object Model (DCOM) or Sun's Java Remote Method Invocation (RMI). Multimedia technologies can take advantage of the portability, location transparency and interoperability that middleware provides to enable efficient, flexible and scalable distributed multimedia applications.

Developing a system that can provide end-to-end real-time and QoS support for multimedia applications in a distributed environment is a difficult task. Distributed multimedia applications are characterized by potentially variable data rates and sensitivity to losses due to the transmission of data between different locations in local- or wide-area networks and the concurrent scheduling of multiple activities with different timing constraints and Quality of Service (QoS) requirements. Several QoS architectures (Aurrecoechea, Campbell & Hauw, 1998) that incorporate QoS parameters (such as response time, jitter, bandwidth) and QoS-driven management mechanisms across architectural layers have emerged in the literature. Examples include the QoS Broker, COMET's Extended Integrated Reference Mode (XRM), the Heidelberg QoS model and the MAESTRO QoS management framework. Providing end-to-end QoS guarantees to distributed multimedia applications requires careful orchestration of the processor resources, as multimedia interactions may lead to excessive utilization and poor quality of service, and multimedia applications can easily suffer quality degradation during a multimedia session caused by network saturation or host congestion. Efficient management of the underlying system resources is therefore essential to allow the system to maximize the utilization of the processors' resources and to adapt to transient changes in the load or in the availability of the resources.

The goals of this chapter are to present a distributed framework for coordinating and managing the delivery of real-time multimedia data. The framework manages the transmission of real-time multimedia data and uses current resource measurements to make efficient management decisions.

CORBA

The Common Object Request Broker Architecture (CORBA) (Object Management Group, 1999) developed by the Object Management Group (OMG) has become a widely accepted commercial standard for distributed object applications. CORBA provides an architecture and platform-independent programming interfaces for portable distributed object computing applications.

The CORBA core includes an Object Request Broker (ORB) which acts as the message bus that provides the seamless interaction between client and server objects. CORBA