QoS as Middleware: Bandwidth Reservation System Design

Gary Hoo and William Johnston, Lawrence Berkeley National Laboratory
Ian Foster and Alain Roy, Argonne National Laboratory and University of Chicago
http://www-itg.lbl.gov/Clipper/QoS/

1. Introduction

We wish to provide quality of service (QoS) to scientists who are remotely controlling experiments on singular instruments such as the LBL Advanced Light Source (ALS) or who need to harness heterogeneous and distributed computing resources to perform large-scale computation. To ensure the timely availability of the computing, storage, and networking resources required for data collection and/or analysis, use of these resources must be scheduled in advance.

The *bandwidth reservation system* (BRS) reserves time in IP differentiated service classes [1]. These classes have upper bounds on the total bandwidth allocation. The reservation unit is a *slot* - a well-defined period of time with an associated bandwidth. The sum of the allocated bandwidths in all of the slot allocations never exceeds the maximum bandwidth defined for the service class. The result is that the classes are never oversubscribed.

2. System overview

2.1. System components

In a distributed computing environment, a client obtains resources in two stages: *reservation* and *claiming*. The client uses a *broker service* to reserve *resources*, such as a service class supported by a router. The broker negotiates with each resource's controller or manager. The bandwidth reservation system (BRS) performs such "local" reservation functions for differentiated network services and ATM QoS, as well as performing access control and resource management during claiming.

The BRS consists of one or more resource managers. Each *resource manager* (RM) manages communication with brokers and other resource managers, and coordinates three other components. The RM consults the *policy decision engine* during reservation to ensure that the requestor is authorized to reserve the resource at all, and during claiming to verify that the requestor is authorized to use a particular reservation. The *slot manager* allocates slots during reservation on behalf of a resource. Finally, the *resource interface module* interacts at claim time with the physical component representing the resource; for example, the resource interface module causes a router to mark a flow for differentiated services treatment.

2.2. Bandwidth reservation

An operational description of the reservation process follows. A client C1 at site 1 asks the broker for premium bandwidth to C2 at site 2. The broker contacts the *first-hop resource manager*—the RM associated with the service provider ingress gateway closest to the client—and requests the reservation. The RM authenticates C1 and checks its access privileges for the premium bandwidth. (Within the Globus environment, the BRS will use the Globus authentication mechanism [3]. To check the requestor's privileges, the BRS will use the Akenti [4] policy-based access control system.) The RM responds to the broker with two types of information: its own, local slot availability, and the identity of the next RM that must be contacted. Each intermediate RM, like the first-hop RM, responds with slot availability and next-hop location. In this way, the broker is guided through the network to each resource that must be reserved. (The scope of an RM in terms of network elements is determined by the ISP, and might represent either a single router or a collection of routers scheduled as a unit; indeed, it might represent the entire ISP domain. In the latter case, the broker and RM mechanism will probably have to exist within the ISP domain, and be mediated through the ingress RM.)

The simplest negotiation for a given bandwidth during a given time period is to request one slot, with the response of all RMs being positive or negative. More sophisticated negotiation would require the managers to respond with a list of slots of the requested duration within some range of the original starting time. With this information the broker could pick, perhaps based on some client criteria, some window when all resources are available.

2.3. Trust between resource managers

Resource managers enjoy bilateral trust relationships with one another. A given RM is configured to know its upstream and downstream neighbor managers and will recognize messages that have been digitally signed by them. The broker passes the token representing an RM's successful reservation to the next-hop RM. The next-hop RM verifies the token's signature and attempts to make the reservation locally. If for some reason the token's signature cannot be validated, the next-hop RM refuses to make the reservation and instead returns an error. The RMs thus form a transitive chain of trust. However, the trust is strictly limited, inasmuch as the reservation token only represents the upstream RM's consent to the reservation. Each downstream RM makes its own, local decision as to whether its resource(s) can be committed.

After obtaining all reservations, the broker presents all of the reservation tokens it obtained to the first-hop RM so that the latter may check the signatures. If all the tokens are verified, the first-hop RM creates a signed *reservation handle* representing the full reservation path, and returns a reservation handle identifier, or *reservation ID*, to the broker for forwarding to the application. The reservation handle itself is stored in a service authorization server (a secure repository for digitally signed documents).

At site 2, the RM must request authorization from the site 2 access control system to use this resource. For this purpose, C2 must have provided C1 with a *proxy certificate*, a digitally signed document declaring that C1 is authorized to reserve resources on C2's behalf. The exchange of the proxy certificate occurs out of band prior to reservation. C1 provides the proxy certificate to the broker during its initial contact. The broker presents the proxy when it requests resource reservation from the site 2 RM. The site 2 resource manage returns its local slot availability but no next-hop information; without a next hop to contact, the broker ends the reservation process.

2.4. Bandwidth use (claiming)

A client claims a reservation by presenting the identity of the claimant (e.g., its own identity as represented by an X.509 identity certificate) and the reservation ID to the first-hop RM. The RM in turn recovers the reservation handle from the service authorization server and checks only its own signature, as the signatures of the internal resource were checked at reservation time. (Other RMs do not need to perform claim-time checking, since the first-hop RM verified that it had a valid reservation covering all internal resources.) The reservation handle is presented to the first-hop RM, which performs any needed *runtime checks*, that is, any checks that could not be performed during reservation. At least one runtime check will probably be required in all cases: the first-hop RM will have to confirm that no topology changes have occurred. If the path has changed and a reservation at a new RM is required, the existing full path reservation is considered unfulfillable. We are investigating how best to recover from such a fault.

If, after validating the reservation handle and making any runtime checks, the first-hop RM finds no problems, it invokes a network management function that performs the required operations to place the flow into the class specified in the reservation. For IP differentiated services, the RM notifies the packet classifier and the flow shaper of the flow identity and characteristics.

3. Current Status

Some of the ideas presented here, such as a prototype slot manager, have been incorporated into a resource management system that supports advance reservation and co-allocation of network and other resources [2]. We are working with Cisco Systems to implement a version of this system on top of its existing RSVP and COPS implementations, and will make use of a differentiated services implementation (also to use COPS) when the latter is available.

We expect this design to evolve with implementation experience.

4. References

[1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss. An Architecture for Differentiated Services. Internet Engineering Task Force Request for Comments 2475.

[2] I. Foster, C. Kesselman, C. Lee, R. Lindell, K. Nahrstedt, A. Roy. "A Distributed Resource Management Architecture that Supports Advance Reservations and Co-Allocation." *Proc. of the Intl. Workshop on Quality of Service, 1999.*

[3] I. Foster, C. Kesselman, G. Tsudik, S. Tuecke. "A Security Architecture for Computational Grids." *Proc. 5th ACM Conf. on Computer and Communications Security*, pg. 83-92, 1998.

[4] M. Thompson, W. Johnston, S. Mudumbai, G. Hoo, K. Jackson, A. Essiari. "Certificate-based Access Control for Widely Distributed Resources." To be presented at the *8th Usenix Security Symposium*, Washington DC, August 23-26, 1999.

This work is supported by the U. S. Dept. of Energy, Office of Science, Office of Advanced Scientific Computing Research, Mathematical, Information, and Computational Sciences office (http://www.er.doe.gov/production/octr/mics), under contract DE-AC03-76SF00098 with the University of California and contract W-31-109-Eng-38 with the University of Chicago. The authors may be contacted at: gjhoo@lbl.gov, wejohnston@lbl.gov, itf@mcs.anl.gov, roy@mcs.anl.gov. This document is report LBNL-42947.