Application Servers
G22.3033-011

Session 4 - Sub-Topic 2
Network Programming with Sockets and RMI

Dr. Jean-Claude Franchitti

New York University
Computer Science Department
Courant Institute of Mathematical Sciences

---

Agenda

- Sockets
- Datagram Sockets
- Simple Serialized Object Protocols
- Remote Method Invocation

---

Java.net Package

- Contains classes for communications and working with network resources
- Two categories of classes:
  - Sockets API
    - All network protocols used between Internet hosts
    - Sockets are the basis for all portable networked communications
    - Remote method calls and distributed objects are implemented over sockets
  - Classes for working with URLs
Sockets

- Low-level programming interface for networked communications
- Send streams of data between distributed applications
- Originated in BSD Unix
- Can be used with almost any kind of underlying network protocol

Java and Sockets

- Java supports a simplified OO interface to sockets
- Network I/O uses the stream classes
- Java provides different kinds of sockets
  - connection-oriented protocol (Socket / TCP)
  - connectionless protocol (DatagramSocket / UDP)
  - multicasting

Clients and Servers

- java.net.Socket & java.net.ServerSocket
- Application acting as a server creates a ServerSocket object and waits
- Server blocks on a call to accept( )
- Upon a connection, accept( ) creates a Socket object
- Server carries multiple conversations at once
- Clients need a host name and port number to connect to server(s)
- See sample server programs
Proxies and Firewalls

- Java has built-in support for the SOCKS service
- SOCKS serves as a proxy server for network connections
- `http.proxySet`, `http.proxyHost`, and `http.proxyPort`
- `%java -Dhttp.proxySet=true -Dhttp.proxyServer=foo.bar.com -Dhttp.proxyPort=1234 MyProgram`

Datagram Sockets

- `java.net.DatagramSocket`
- Uses the UDP protocol
- No guarantee that the data will get through
- No guarantee that the data will arrive in the right order

Simple Serialized Object Protocol

- Serialization can pack/unpack entire graphs of interconnected objects
Remote Method Invocation

- Uses object serialization
- Not limited to data structures like RPC
- Can ship the state of an object over the wire and let the recipient interact with the object after receiving it
- Breakthrough of RMI: makes it possible to ship both code and data around the Net

Remote and Non-Remote Objects

- Objects used with RMI must be serializable
- Non-remote objects passed over the network are simply copied
- Remote objects are passed by reference

RMI Registry

- Used to look up a reference to a registered remote object on another host
- Implemented by a class called Naming and an application called rmiregistry
- rmiregistry must be running before you start a Java program which uses the registry
- RMI/IIOP uses JNDI as the naming service (tnameserv)
Alternatives to RMI

- CORBA
- DCOM