CS843 Presentation

Implementation of a Distributed Document-Based System

David Dellinger
Abstract

Distributed document-based systems in use today
  ▶ World Wide Web
  ▶ Commercial software products (Ex: Lotus Notes)

Users need to find information
  ▶ Doesn’t know where the data is located
  ▶ Doesn’t know how the data is accessed
  ▶ May not care that other users can access the same data

Concept is in use by businesses today
  ▶ Company intranets
  ▶ A user-level applications provides retrieval mechanisms
  ▶ Reduced user training costs
  ▶ Provides a consistent, repeatable, modular method of extracting information upon request.
Demonstration Goals

Key Application Facets

Demonstrate the ability to provide a flexible, modular solution regardless of the means of data access.

Documents are stored in different repositories throughout a network of computers.

Documents are retrieved via Remote Method Invocation (RMI) and Common Object Request Broker Architecture (CORBA) technology (other means available).
Application Limitations

Demonstration is Proof-of-Concept Prototype

Limited Scope

▶ Application is intended as technology demonstrator
▶ List of files is impractical in real world use
  - Repositories must include databases, flat filesystems, legacy systems, etc.
  - Search engine would be beneficial
▶ Document types supported is inadequate
Application Features

Flexibility, Modularity

Files can be replicated among servers if desired

Files can be relocated during client execution

Local caching of files used

New document types can be implemented

New access methods (sockets, RPC) can be implemented
Application Features

Administrator’s Role

Determines which files are accessible by users
Determines file location
  ▶ May place files on one or more servers
Determines server pool (how many servers to use)
Determines server communication methods, ports
  ▶ RMI
  ▶ CORBA
  ▶ Other
Stores all settings in application configuration file
  ▶ Changes made without recompiling code
Client

Client is configured at runtime
  ▶ Reads configuration file
  ▶ Establishes server list
    – Server name, communication method and port
  ▶ Establishes file list

Requests selected file from servers until file is returned
  ▶ Loops through server list

Places file from server in local cache, then displays it
Application Features

Servers

Server waits for client request
- Searches for file upon request by client
  - Does not know about files in advance
  - Returns immediately if not available
- Compares file size and modification date with client’s cached copy
  - Returns file if different
  - Allows use of cached copy if same
- Startup script determines server type
  - Configuration file searched for server hostname
  - Server type and port retrieved
  - Matching server program started
Application Configuration

Servers, ports and files are all separated from the code. Changes can be made without recompiling.

RMIPort|50168
CORBAPort|50169
Server|spock|RMI
Server|kirk|CORBA
Server|sisko|RMI
File|/tmp/2002_Midterm_Exam.pdf
File|/tmp/2002_Final_Exam.pdf
File|/tmp/2003_Midterm_Exam.pdf
File|/tmp/2003_Final_Exam.pdf
File|/tmp/CS843_Syllabus.pdf
File|/tmp/2005_assign1.txt
File|/tmp/2005_assign2.txt
File|/tmp/2005_hwk1.txt
File|/tmp/2005_hwk2.txt
File|/tmp/2005_hwk3.txt
File|/tmp/2005_Midterm_Exam.pdf
Client Interface

Transparent File Locations

![Image of a software interface showing file list and file display]
Client Interface

Text file selected

CS 843 Distributed Computing Systems – Homework 1 – Name: ______________

Assigned: Wednesday, February 2
Due: Wednesday, February 9 (in class)

Please show all work on a separate sheet attached to this sheet.
For each question write key points.
(50 points)

1. (2 pts.) What is a distributed system?
2. (4 pts.) What are the goals to build a distributed system?
3. (4 pts.) Name two operating systems for distributed systems and briefly explain what they are.
Client Interface

PDF file selected
Questions?
Acknowledgements


UGS Teamcenter

Dassault Systemes ENOVIA