Distributed Network Management Using SNMP, Java, WWW and CORBA

André Marcheto

Augusto Hack

Augusto Pacheco

Augusto Verzbickas

ADMINISTRATION AND MANAGEMENT OF COMPUTER NETWORKS - INE5619

Federal University of Santa Catarina

OCTOBER 2ND, BRAZIL

Distributed Network Management Using SNMP, Java, WWW and CORBA

(Based on the reference – Distributed network management using SNMP, Java, WWW and CORBA. AM Barotto, A de Souza, CB Westphall. Journal of Network and Systems Management. 8 (4), 483-497, 2000.)

Outline

1. ABSTRACT
2. INTRODUCTION
3. BACKGROUND
   - 3.1. SNMP
   - 3.2. ATM and FDDI
1. DEVELOPMENT ENVIRONMENT

Outline

5. CORBA OVERVIEW
    5.1. ORB
    5.2. GIOP and IIOP
    5.3. NETWORK MANAGEMENT USING CORBA AND WWW
6. LOGICAL STRUCTURE

Outline

1. ABSTRACT

This presentation describes:

- the design and implementation of a Distributed Network Management Using SNMP, Java, WWW and CORBA;

2. INTRODUCTION

The intent of this presentation is to:

- Provide insight about how the technologies works and how they can be used to develop the distributed system.
- Propose a model of distributed system for managing a cluster using the browser to transfer information.

3. BACKGROUND

3.1. SNMP - Simple Network Management Protocol

- Protocol widely used in the management of computer networks.
- Enables administrators to resolve network problems

3. BACKGROUND

3.1. SNMP - Simple Network Management Protocol

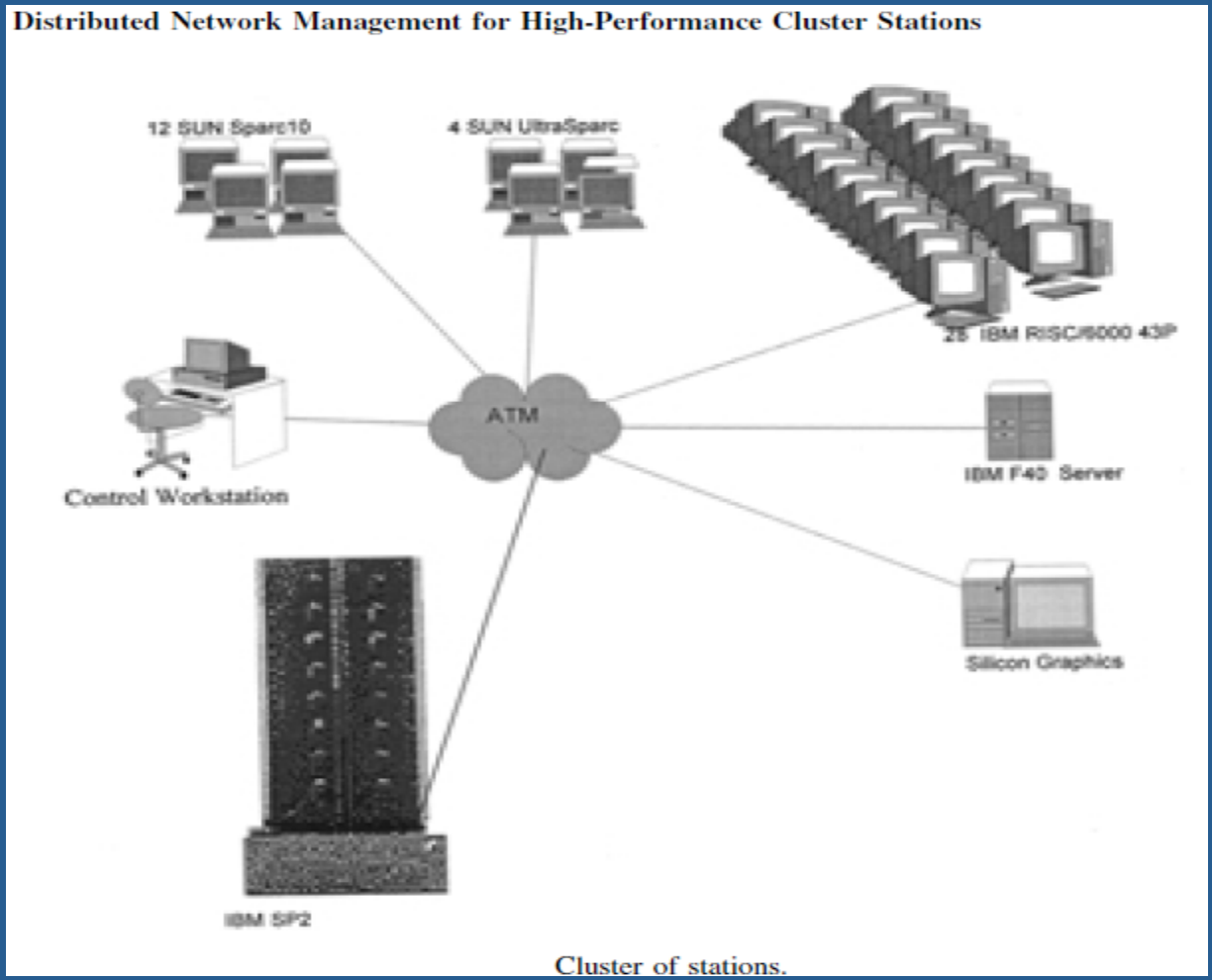
- SNMP managed networks typically consist of:
  - Managed Device
  - Agent - software running on managed devices
  - Network management system (NMS) - software running on the manager

3. BACKGROUND

3.2. ATM and FDDI

- ATM: Asynchronous Transfer Mode - Its a high speed networking standard that supports voice and data.
- FDDI: Fiber Distributed Data Interface - provides a high speed data transfer in local networks.

4. DEVELOPMENT ENVIRONMENT



4. DEVELOPMENT ENVIRONMENT

- Cluster contents:

- IBM 43P workstations
- SUN UltraSparc (SUN processor)
- IBM F40 (Workgroup server)
- IBM SP2 Server

5. CORBA OVERVIEW

Common Object Request Broker Architecture

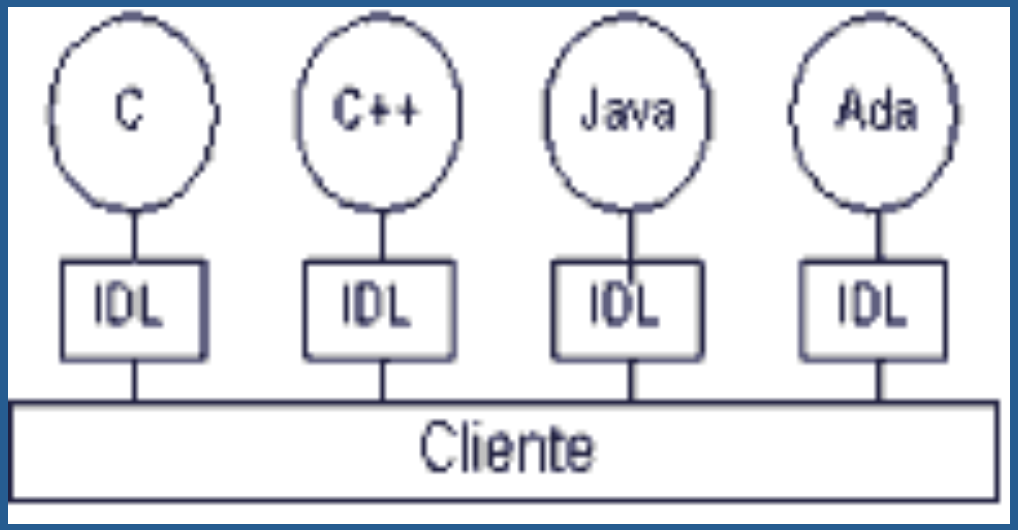
5. CORBA OVERVIEW

- Created by OMG in 1990, CORBA supports the communication between objects placed in any device throughout the network.

- The objects can be implemented by distinct programming languages and reside in different operational systems.

5. CORBA OVERVIEW

- Uses the IDL (Interface Definition Language) to define the interface of the objects.



5. CORBA OVERVIEW

- Through the IDL, the object provides the services that can be used and what should be expected in response.

- The interfaces have to be only declarative and cover the entire operation of the code.

5. CORBA OVERVIEW

- Supported Languages: Ada, SmallTalk, Java, C ++, C and COBOL.

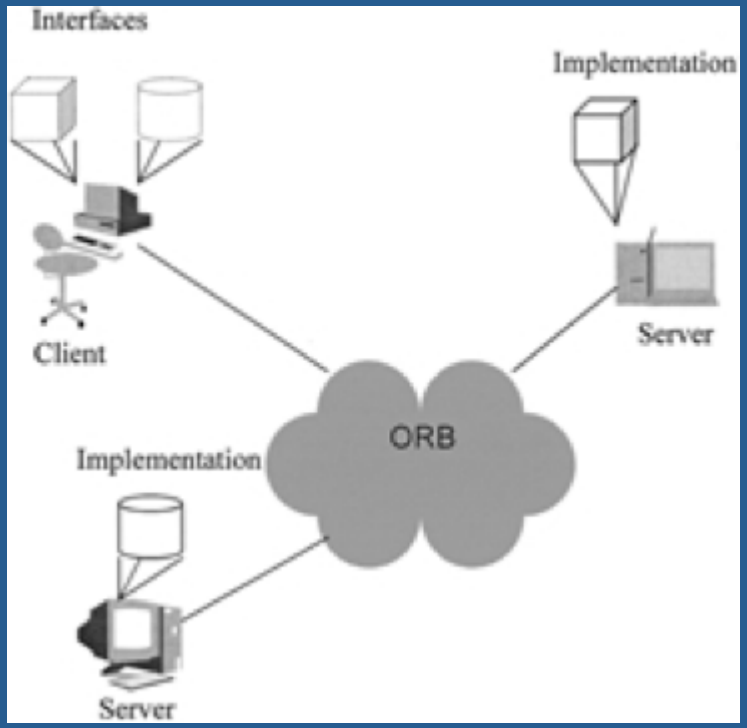
- Supported variables: long, short, unsigned long & short, float, chat, boolean, octet, any, struct, union, enum, sequence, string, array and typedefs.

5.1. ORB

- Object Request Broker
  - Responsible for distribution of messages among objects
  - Enables a client to invoke methods from a local or remote server
  - Seeks the object that has implemented the method
  - No need for the client to be aware of the operating system or where the server is located

5.1. ORB

- The only information required is the interface with the object



5.2. GIOP

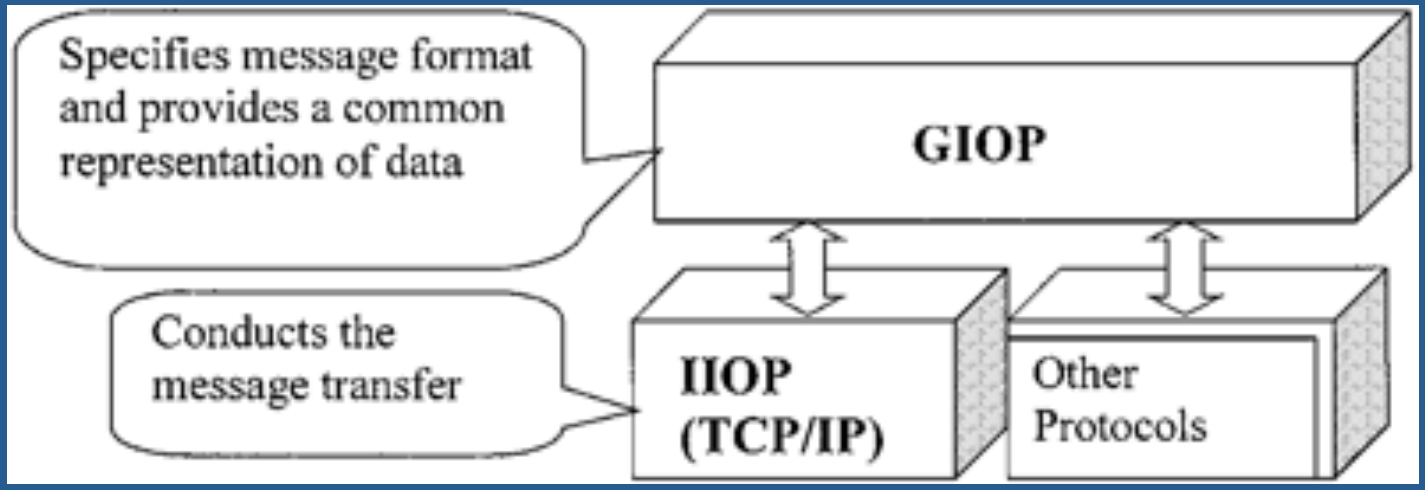
- General Inter-ORB Protocol
- Specifies message format and provides a common representation of data
- Uses the Common Data Representation (CDR)
- Makes possible the interoperability among data types and hardware and operating system independence

5.2. IIOP

- Internet Inter-ORB Protocol
- Defines how GIOP messages are transmited over TCP/IP protocol
- Provides the means to use the Internet as communication mecanism between ORBs
- Allows browsers and servers to transmit integers, arrays and more complex objects

5.2. GIOP and IIOP

- GIOP and IIOP work together to make the interoperability between ORBs possible



5.3. Network Management using CORBA and WWW

- CORBA is being increasily adopted in network management
- main characteristic of CORBA is separating interface from implementation
- makes possible to instantiate objects in separated equipments
- great characteristics for the development of management application

5.3. Network Management using CORBA and WWW

- Management applications are usually complex and require large amount of CPU and memory
- These applications must also have security mechanisms
- The development of complex management applications using Web Technology becomes restrict if there is no support of a standard architecture, as in the case of CORBA.

5.3. Network Management using CORBA and WWW

- Main factors that limit the use of the Web Technology without CORBA are:

- Slow information transfer to the browser
- The mechanisms used to access Web pages, when compared to a server dedicated to management activities, reveal a lower performance
- Lower security

5.3. Network Management using CORBA and WWW

- To obtain a better performance, management applications may be set to a high-performance server
  - The object may be implemented in any language
  - The user of the application needs to know only the definition of the interface
  - The application may run on a single or several servers

5.3. Network Management using CORBA and WWW

- CORBA characteristics result in the following benefits:

- reduction of the number and size of classes that are transferred to the Web Browser
- the client runs only the interface with the user (better performance)
- the client is not aware of the implementation details (better security)
- application is not affected by applet security restrictions

6. LOGICAL STRUCTURE

Instantiate the computer object

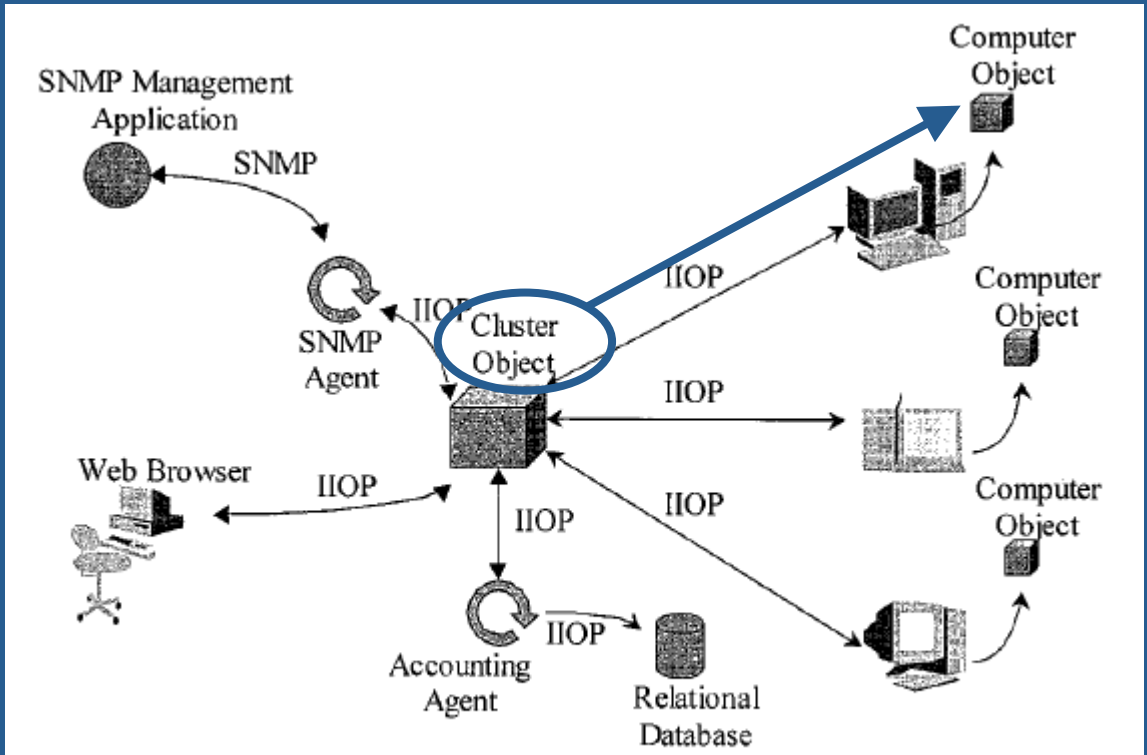


Fig. 4. Logical structure of the *Cluster* management system.

6. LOGICAL STRUCTURE

Cluster Object and Computer Objects communicate through Corba messages

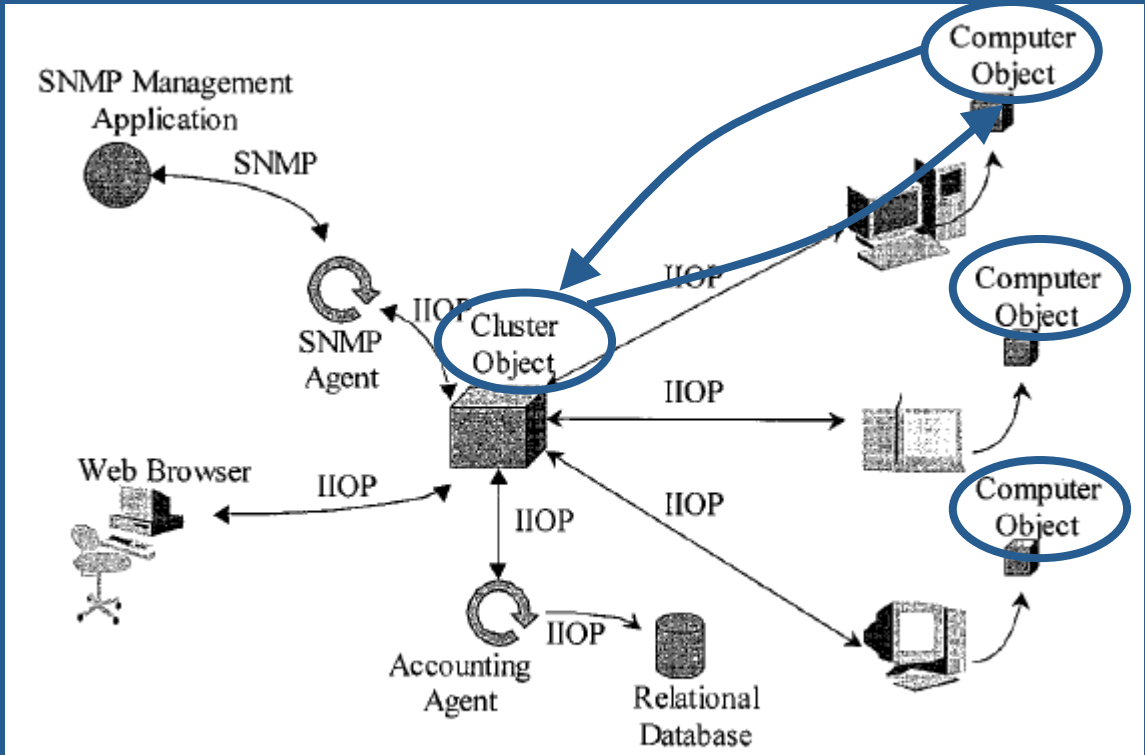


Fig. 4. Logical structure of the *Cluster* management system.

6. LOGICAL STRUCTURE

Information is stored in a relational database by the agent

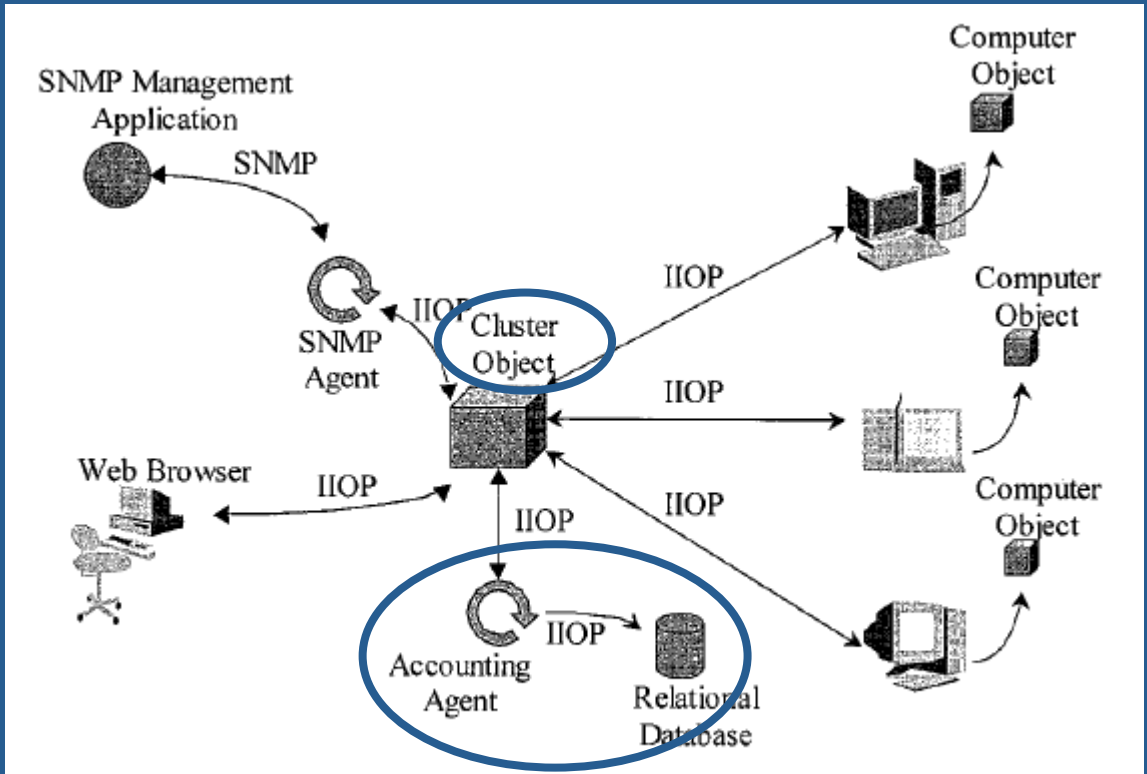


Fig. 4. Logical structure of the *Cluster* management system.

6. LOGICAL STRUCTURE

External SNMP applications communicate with the cluster through a SNMP gateway

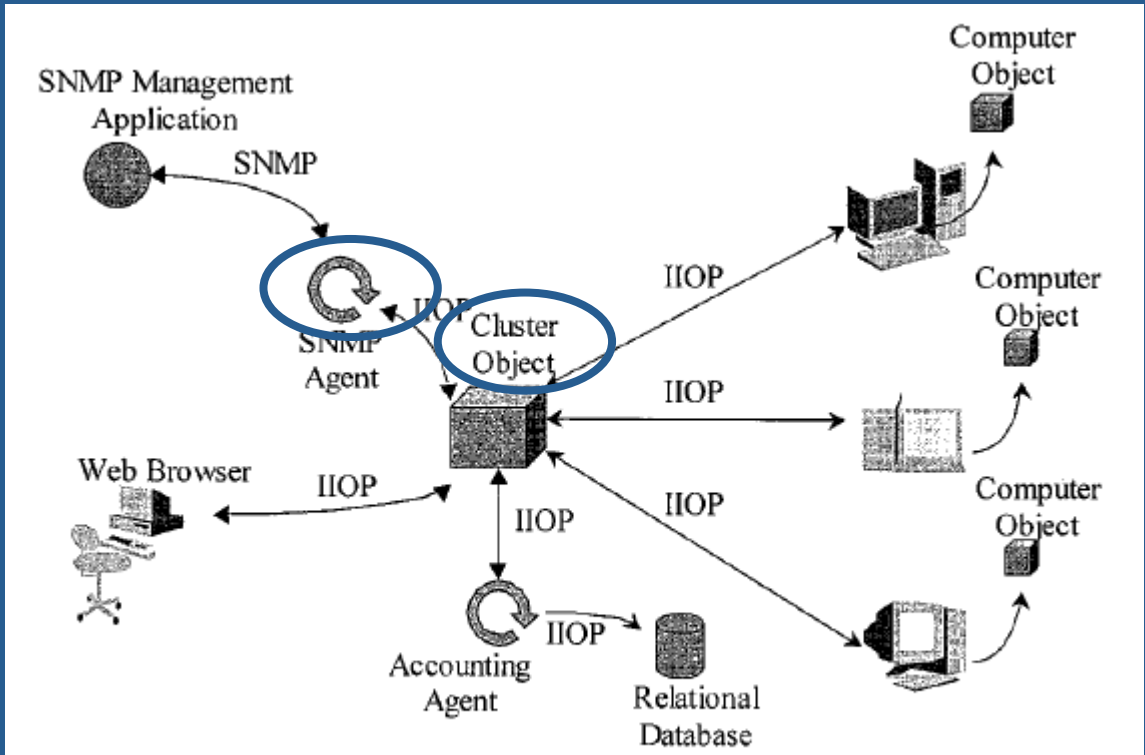


Fig. 4. Logical structure of the *Cluster* management system.

6. LOGICAL STRUCTURE

```
1
2
3
4
5
6
7
8
```

```
package agent;
import java.lang.Object;


public interface Computer
{
    public long GetCPU();
    public String getSystemName();
}
```

6. LOGICAL STRUCTURE

```
1  String hostURL = new String("IIOP:/ / serverName[:serverPort]");
   Computer computer = null;
2
   ORB client = CORBA.ORB_init(null);
3  computer=(Computer)client.URLToObject(hostURL+";/ computer?agent.Computer")
   System.out.println("system " + computer.getSystemName());
4  client.close();
5
6
7
   > java COM.ibm.corba.IIOP.IIOPSServer
8
```

7. ACCESS TO A RELATIONAL DATABASE

- Storing management information in a relational database
- Construction of applications, written in any programming language, to invoke SQL commands in order to handle the data and provide better means for analyzing the behavior of the Cluster.
- JDBC - Java Database Connectivity

8. IMPLEMENTATION OF THE MANAGEMENT APPLICATION

- Development of an applet for the collection of CPU usage rates from all the equipment in a cluster to identify the equipment containing the lowest CPU usage rate.
- A specific agent collects information from the cluster and stores this information in a relational database.

9. Integration with other management applications

- For the integration of the system with other management applications, we have developed an agent to act as a gateway between the SNMP and the IIOP protocols

- This agent consists of an application that monitors UDP ports

9.1 Agent tasks

- When the agent receives an SNMP get request, the following tasks will be performed:
  – identification of the SNMP parameter being required
  – identification of the requested operation and instantiation of the object cluster for the proper collection of management information
  – when the response is received, a get-response PDU is returned

10. Conclusions

- The prototype implemented in this project defines a simplified management structure in which the managed objects provide information pertinent to the system and the CPU of the stations within the cluster
- A user may access management information from any machine connected to the network using a Web browser

10. Conclusions

- Management and instantiating mechanism of objects portable to any platform which has a Java interpreter

- Java Runtime class allow the instantiated objects (using CORBA) to invoke commands from a Unix shell

10. Conclusions

- Management system functions that allow a user, through interaction with the Java applet, to perform tasks such as:
  – automatic installation of software in all the cluster stations
  – removal of outdated log files and management of disk space

10.1 Improve

- Security mechanisms for the management of the cluster

- Only map into the SNMP protocol the functions which show a significant relevance to the management of the system because it is limited compared with CORBA.

11. REFERENCES

References indicated in this presentation:

- **Distributed network management using SNMP, Java, WWW and CORBA. AM Barotto, A de Souza, CB Westphall. Journal of Network and Systems Management. 8 (4), 483-497, 2000.**