A Comparison of Java RMI, CORBA, and Web Services Technologies for Distributed SIP Applications

Mark D. Hanes   Stanley C. Ahalt   Ashok K. Krishnamurthy

Department of Electrical Engineering
The Ohio State University

Poster C.2
September 26, 2002
High Performance Embedded Computing Workshop (HPEC 2002)

Motivation & Research Goals

- To promote more rapid development of more easily-maintained SIP (Signal and Image Processing) software applications through the use of middleware standards.
- To make effective use of legacy code and existing applications whenever possible.
- To make use of established network protocols to ease the burden on programmers and to facilitate code re-use.
- To make use of emerging discovery and service-oriented paradigms for distributed computing applications.
- To compare and contrast current and emerging middleware technologies for distributed computing SIP applications.

Middleware Technologies

- Early distributed computing models focused on a model of remote procedure calls.
- Current focus is on ‘remote objects’ and their use.
- Emerging web services are built on ‘messaging’ concepts, which frequently take the form of request/response method calls on remote objects.
- The technologies establish well-defined protocols for communication between computing elements.
- Depending on technology, language-independence and platform-independence are available.
- Many middleware technologies provide ‘discovery’ for use in defining and providing services.
- Our current focus: Java RMI, CORBA, Web Services (SOAP/XML)

Java RMI (Remote Method Invocation)

- A server object (an instance of a class) exposes one or more interfaces to potential clients.
- Server object registers itself with a simple form of discovery service to provide access to its services.
- Language-specific, in that interfaces are written in Java.
- Platform independent as a result of Java’s platform independence.
- Provides programmer-transparent conversion of method calls to remote method calls.
- Supports both JRMP and IIOP (from CORBA) as ‘wire-protocols’ for method calls.
- Any platform that interacts with or supports Java can take advantage of this technology (*e.g.,* Matlab).

CORBA

Common Object Request Broker Architecture

- Language independent, in that a variety of programming languages are supported.
- Exposed service interfaces are described in language-neutral IDL.
- Well-suited for integration with legacy code and applications.
- Communicates using standardized IIOP (Internet Inter-Orb Protocol).

Web Services

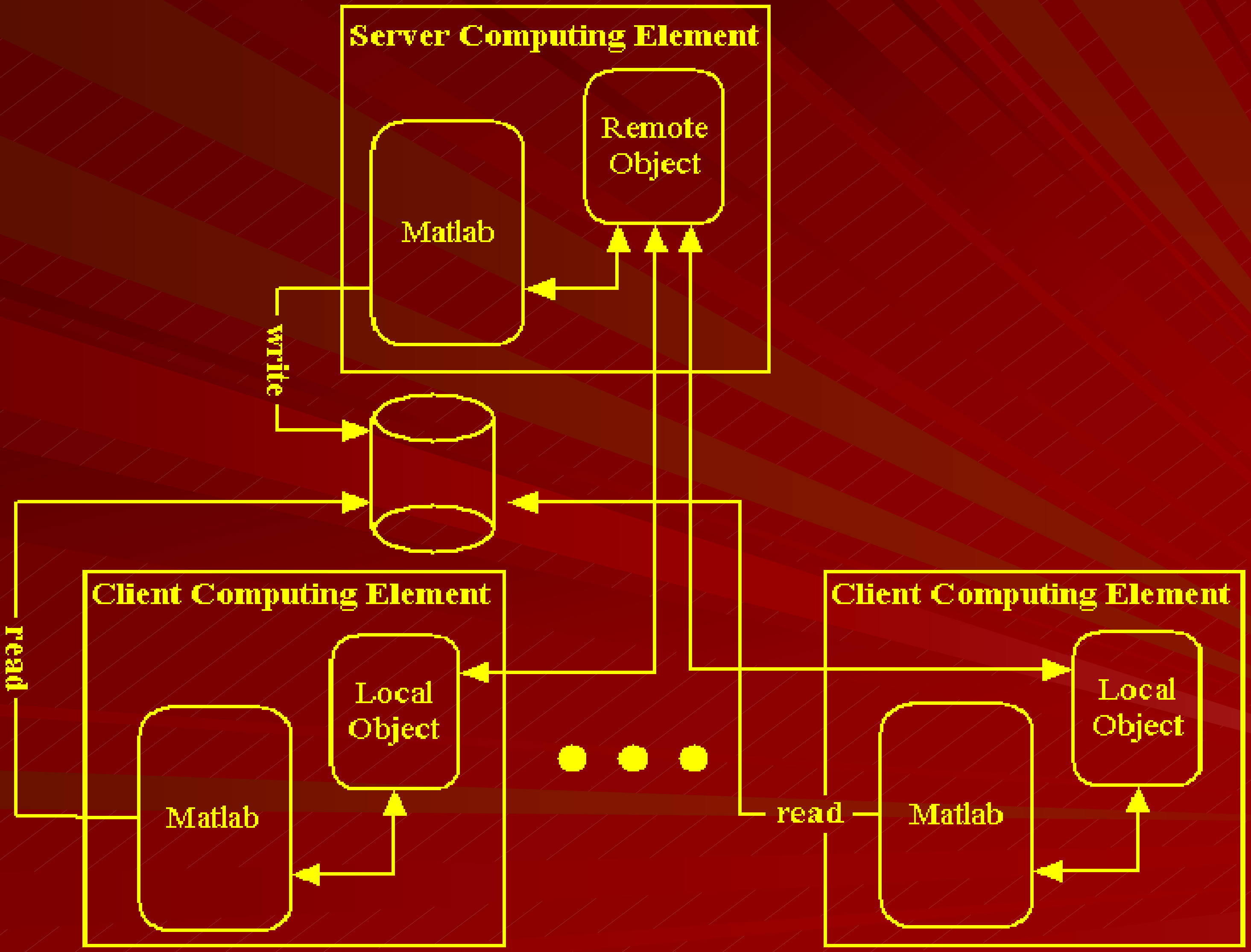
SOAP/XML

- SOAP protocol (XML-based) used to describe messages passed across a network.
- Messages carried on a network protocol such as HTTP, HTTPS, SMTP, *et al*.
- Interfaces are language independent, in much the same way as CORBA. Interfaces are described in WSDL.

Example: Clustering Algorithm

- Goal: Find a specific number of cluster centers in a supplied data set.
- Distributed algorithm written in Matlab.
- Inputs:
  - Data set:
    - *M* – dimension of each data point
    - *N* – Number of data points
  - Centers:
    - *c* – Number of centers to locate in data set
  - Computing elements:
    - *L* – Number of available computing elements
- Outputs:
  - The *c* located cluster centers, each of which is dimension *M*.

An RMI Client-Server Architecture for Distributed Matlab Applications



RMI Architecture Communications

- Server Matlab process initializes Java server-object and exports interfaces to the internet.
- Server Matlab process initiates client startup, providing instructions for contacting remote server object through a common file system.
- Server Matlab process allocates data to each client and exports relevant data to the server object.
- Server Matlab process alerts clients, through the remote object, that data is ready.
- Server Matlab process, meanwhile, waits for client processes to complete task.
- Client Matlab processes retrieve data, process, and report results back to server remote object.
- Client Matlab processes await instructions to terminate or process more data.
- Server Matlab process collects data, analyzes, and then repeats or finishes. Server notifies clients, through server object, of decision.

Future Work & References

- Implement CORBA and SOAP/XML architecture for Matlab-oriented, distributed SIP applications.
- Conduct performance comparisons of middleware technologies.
- Generalize framework for distributed SIP applications.

- References

  - Brose, Noffke, and Muller, *JacORB 1.3 Programming Guide*, www.jacorb.org, 2001.
  - Brose, Vogel, and Duddy, *Java Programming with CORBA*, Wiley Computer Publishing, 2001.
  - Graham et al., *Building Web Services with Java*, Sams Publishing, 2002.
  - Seshadri, *Enterprise Java Computing: Applications and Architecture*, Cambridge University Press, 1999.
  - Snell, Tidwell, and Kulchenko, *Programming Web Services with SOAP*, O'Reilly, 2002.