Jabber Grid:
A Distributed Problem Solving System.

CIS 630 Project
Dan Keith, Kim Griggs, Mustafa Hakim
JabberGrid: Introduction

• **Distributed problem solving systems (DPS):**
  – concerned with how large-scale problems can be solved using systems of agents, distributed among a set of networked computers, working cooperatively.
  – **Example:** Seti@home

• **DPS approaches can potentially offer increased processing power, flexibility, and reliability, and reduced hardware costs.**
JabberGrid: *Introduction*

- **Focus of the JabberGrid project:**
  - *Using the Jabber/XMPP (Extensible Messaging and Presence Protocol) instant messaging technology as a communications and coordination platform.*
  - *Potential of JabberGrid as a resource broker in a collection of distributed and heterogeneous resources (Grid Computing).*

- **Developed a small set of demonstration applications for JabberGrid that solve some simple problems in parallel on several machines.**
JabberGrid: Jabber Technology

• The Jabber Project:
  – An open-source alternative to Instant Messaging technology provided by AIM (AOL Instant Messenger) and YahooMessenger.

• Jabber technology provides:
  – Store-and-forward capability
  – Security and Fault Tolerance
  – Extensible messaging system (XMPP).
  – Ability for diverse programs to use the network for a variety of purposes
JabberGrid: JabberGrid Architecture

Workers, Tasks, and IM clients all share access to the same XMPP Network "cloud".
JabberGrid: Distributed Work

- **The 'WordWorker' Task**
  - text processing task
  - entails many important problems in distributed environments and systems such as; action coordination, task scheduling and resource allocation

- **The 'PingPong' Task**
  - workflow task

- **The 'WordWorkerPool' Task**
  - hiring pool task
    - matchmaking services
JabberGrid: Challenges

A number of resources willing to perform some work/provide service. A client/another service wants to use these resources to perform some work.

What are the problems?

1. **Resource Discovery.**

Method - I

- Contact a central authority which will return a list of available resources. (Can be very large)
- Then the client requesting the service has to select based on some criterion.

Method - II (Implemented)

- The Client knows a channel (Room in our implementation) on which all resource providers are listening.
- Client Multicasts the request on the channel.
- Eligible resources report to the client requesting the service.
2. **Co-Ordination.**

Co-ordination problem in hiring
- How to make sure only the required number of resources report for work to the client?

Co-ordination problem in work distribution
- How to make sure that all resources do not carry out the same work?

**Method - I (Single Server Approach)**
- Each resource handshakes with the client
- Client decides who gets the work.

**Method - II (Distributed Approach)**
- Smart Multicasting (Implemented)

**Advantages:**
- Co-ordination Problem is solved.
- System remains transparent.
JabberGrid: Challenge

• **Workflow**
  – offers reliability, scalability, and efficient load distribution in problem solving.

• **In JabberGrid:**
  – *Workflow management can be coordinated through the point-to-point communication or the smart multicast.*
JabberGrid: Investigation

• **Fault Tolerance**
  – *In Jabber no single server handles the communication for all of the workers.*
    • *Problem: A worker claims work then crashes*
      – *How does a DPS guarantee returns?*
        » *Persistence in Jabber*

• **Security**
  – *Jabber technology provides security measures.*
    • *All Jabber servers are isolated from the public Jabber network, and robust security has been built into the core XMPP specifications.*
JabberGrid: Investigation

• **Problem Suitability**
  – *Problems where the worker needs to spend moderate to large amounts of time to generate a result.*

• **Recursive problems**
  – *The workers that process the work must synchronize with their children.*
    • Workflow: Where each workflow step is an instruction.

• **Monte Carlo Simulations**
  – *Task is completed by distributing the work of simulating random variables to multiple workers.*
    • Each worker can do its work without synchronizing with any other worker.
JabberGrid: Evaluations

• The metrics used on the JabberGrid project:
  – Performance issues such as bandwidth
  – Throughput time it takes to solve problems
  – Size of work packets before experiencing a bottleneck.

• Benchmarks:
  – Empty Payload
  – Big Payload (1 megabyte)
  – Large Payload (2 megabytes)
JabberGrid: Performance

• Compared JabberGrid to Java RMI
  – the speed we were interested in was not raw throughput, but how long it took to do the work.

• Test Case:
  – WordWorker: Ping-Pong Task
    • Ping: Receive Data
    • Uppercase Data
    • Pong: Send data back
JabberGrid: Performance

- Initial Performance Results
JabberGrid: Evaluations

• **Scalability**
  – *metrics: throughput and packet size.*

• **How large of problem can JabberGrid solve?**
  – *Relationship between the payload consumed/generated by the worker and the amount of time that the worker needs to solve the subtask.*
JabberGrid: Scalability

- **Throughput times and packet size**

![Packet Size Chart](chart.png)
JabberGrid: Future Work

- Enhance workflow packets
- Implement presence to use as a discovery services.
- Test Server-to-Server Communication.
- Build more interesting workers – translators, stock ticker, etc..
- Build a distributed game using JabberGrid as communication architecture.
- Compare GRID specification in OGS to JabberGrid.
JabberGrid: Conclusion

• The goal of the project was to show that Jabber technology has the potential of being an important keystone technology in grid computing.

• JabberGrid demonstrates this potential by using Jabber to implement a distributed problem-solving system.