

# *Access Grid Recording/Archiving Services*

Joseph Curtis

Joe (Xianzheng Zhou)

Quoc Le

Rehan Walsh

Viral Patel

Chao Yan

Alekh Aggarwal

---

---

# Access Grid

- Large scale video-conferencing.
  - High bandwidth (uses UDP/IP and RTP for video and audio, TCP/IP for application sharing)
  - Multicast Networking
  - Used for large scale distributed meetings, collaborative work sessions, seminars, lectures, etc...
  - Consists of many nodes (usually a room like this one)
  - Large screens, video and audio from many nodes at once
- 
-

# *The Problem*

- Recording Access Grid sessions
  - Many video and audio streams
  - High bandwidth, lots of data
  - Needs robust file format (with indexing for fast skipping)
  - Edit Access Grid sessions
- 
-

## *The Problem (continued)*

- Perform basic editing functions
- Remove stream
- Crop stream
- Convert the recorded sessions to standard video formats

# *Original System*

- Prototype Remora server written in Java
  - Primitive telnet-based interface
  - Protocol for controlling Remora server
  - Remora file format
  - No client
  - No Post Production editing tool
- 
-

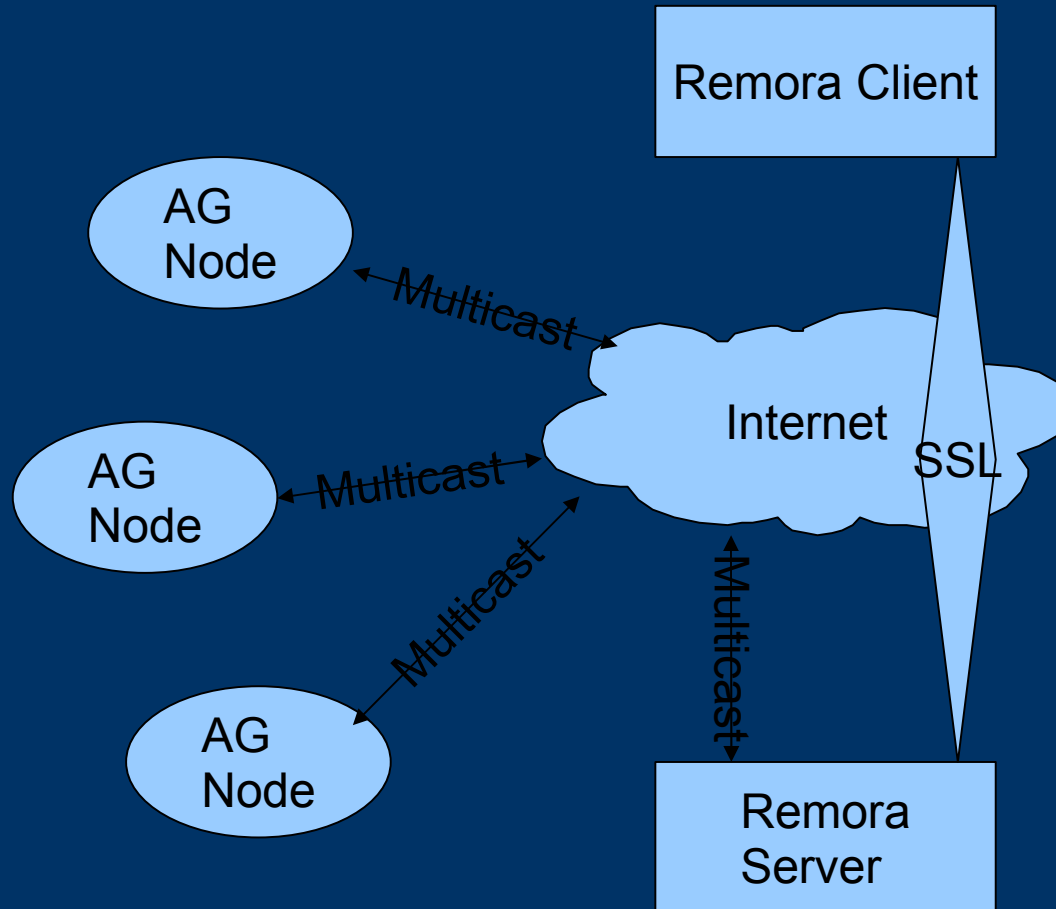
# *Original System Problems*

- Intended to be a prototype
  - Written in Java (Client wanted it written in C)
  - No security implemented in Remora Server
  - No Client to interface with the server
- 
-

# *New System*

- Implement Remora Server in C
  - Improve on the original prototype
  - Use the same indexed file format as the original prototype
  - Implement a Client application to interface with the Server
  - Implement a Post Production tool for performing basic editing function
  - Convert Remora file format to standard formats...?
- 
-

# Basic Structure of Our Solution



# *Key Deliverables*

- Operational Concept Document
  - Software Development Plan (with WBS)
  - Software Requirements Specifications (Remora and Post Production)
  - Software Design Documents (Remora and Post Production)
  - Implementation (Remora Server, Remora Client and Post Production)
  - Test Plans (Remora and Post Production)
  - Test Reports (Remora and Post Production)
- 
-

# *Key Requirements*

- Remora server written in C
  - Remora remote client implemented (Partially Satisfied)
  - Remora server and client communicate via SSL
  - Server records and plays back Access Grid sessions in the file format (Partially Satisfied)
  - Post Production editing tool implemented and compatible with same file format
- 
-

# *Issues Faced*

- Lax performance during first half of the year
  - Inadequate planning
  - Some team members lack of understanding of the nature of the problem
  - Lack of communication between team members
  - Team member absent during most of second half of the year
  - Technical difficulties
- 
-

# *Product Status (Joe)*

- Remora
    - Remora Server
    - Remora Remote Control Client
    - Post-production Tool
  - Improvements Needed
    - More friendly interfaces
    - Server need better error tolerance capability
    - User Manual
- 
-

# *Tasks Participated (Joe)*

- Management:
    - Project planning
    - Maintaining WBS
    - Scheduling tasks
    - Supervise task status
    - Packaging
  - Development:
    - SRS
    - SDD
- 
-

# *Tasks Participated Cont. (Joe)*

- Development Cont.
  - Implementation
  - Testing/debugging



# *Lessons learned (Joe)*

- Experienced technical problem that can seriously delay the project schedule
  - Why more precise status measures are needed for project management
- 
-

# Tasks for Chao Yan

- My primary responsibilities part of the Access Grid team were to implement the Post Production Tool, including all the documents and coding.
    - Software Requirement Specification: I spent 15 hours writing and reviewing SRS.
    - Design and prototyping: I spent 128 hours on this part. It consists of the following 5 sub-tasks. I did not spend hours on “Search and Learn MPEG library 3” because it is too complicated and we do not have enough time for that so we have decided not to do it.
- 
-

# Tasks for Chao Yan

- Post Production design document: (56 hours 45 minutes)
    - This part includes both high level design and the detailed design document.
  - Post Production design Prototyping: (36 hours 30 minutes)
    - We started coding early to make sure our design decision does work.
  - Documenting Source codes: (12 hours 45 minutes)
    - We decided to use JavaDoc style to document our source codes. Also, I have changed some class, variables names that make little sense.
  - Post Production SDD re-documenting: (22 hours)
    - After we have made some changes to the source codes, we need to make some minor changes in the document to make it consistent with the source codes.
  - Search and learn MPEG library 3 (0 hour)
    - Not implemented.
- 
-

# *Lessons Learned (Chao Yan)*

- Post Production Tool Implementation: I spend 23 hours and 15 minutes on it. This is the actual coding part for post production tool and it is based on the codes from prototyping sub-task.
- Issues faced:
  - Technical unfamiliarity:
    - I am not familiar with the network packet structures (for example, the RTP, RTCP, UDP packet).
    - The reference document was hard to understand.



# *Lessons Learned (Chao Yan)*

- Things that went right:
    - This project is a good chance to practice what I have learned in the previous year.
    - I also gained knowledge and understanding of some software technologies, especially Java.
  - Things that went wrong:
    - I did not spend too much time on the project in the first semester. It made me hard to catch up the hours.
- 
-

# *Project objectives (Rehan Walsh)*

- What has been discussed previously
  - Performance
  - Standards
  - Extensions
- A grander motivation – AG accessibility
  - Functionality
  - Useability



# *Work expected, Work done (Rehan Walsh)*

- Initially coding
    - Overshadowed by implementation
  - Then it was about documentation
  - OCD, SRSs, STP, STD, STR, SDD Review
  - Coding and bug fixing – Remora C, ClientApp
  - PPT Testing
- 
-

# *MIL-STD-498 (Rehan Walsh)*

- All our documents are based around this standard
- Why
  - Coverage
  - Plain English
  - Tailoring

# *Operational Concept Description (Rehan Walsh)*

- An important document
  - “describes a proposed system in terms of the user needs it will fulfil, its relationship to existing systems or procedures, and the ways it will be used” – MIL-STD-498
  - First step in the project
  - For us, a changing document
    - Technical feasibility
    - Time constraints
  - For the developers after us
- 
-

# Remora and PPT SRSs (Rehan Walsh)

- Where I spent most my time
  - “Requirements for a [CSCI] and the methods to be used to ensure that each requirement has been met” – MIL-STD-498
  - Check-list of features, Testing requirements
  - Merging SRSs
    - PPT, Remora Server, Remora Client →
    - PPT, Remora
  - Living documents
- 
-

# *PPT Test Plan and Testing (Rehan Walsh)*

- MIL-STD-498 distinction
    - STP
    - STD
  - Client verification (acceptance)
  - Successful here – bug identified
  - Reporting bugs vs speculating on fixes
- 
-

# *Issues faced (Rehan Walsh)*

- Teamwork
  - Hardest aspect of the project
  - Communication
- Consensus
  - Not always the best approach
- Changing requirements
  - Common goals continued



# *Lessons learned (Rehan Walsh)*

- How to develop an application!
  - Proper time management
    - newbie problems?
  - Get docs done first
    - Project roadmap
    - Project boundaries
  - Put administrative structures in place
    - Communications, file storage
    - Promotes efficiency
    - Provides support
- 
-

# *Thanks! (Rehan Walsh)*

- 3100 “added a lot of value” to me
- Best course in which to learn real world skills



# *Conclusions*

- Performance lax at start of project
  - Pulled it together in the second half
  - Delivered value to the client
  - Our work can be built upon
- 
-

## *Next Steps...?*

- The server (and file format) need to be improved to solve the byte order problem
  - The server also needs to be more tolerant of invalid states
  - The client needs a more friendly user interface...
  - So does the post production
  - Implement the video conversion that we scoped out
- 
-

*Questions?*

