**Interactive Visualization**

- "Active Visualization"
- Additional modifications by Fangzhe Chang

---

**Structure of the Application**

![Diagram of server and client communication](Diagram.png)

**Additional Details: Server**

- **Main thread**
  - waits for messages from client
  - if (message := VIEW_NEW_IMAGE)
    - set up data structures that track progress of this download
    - send client "header info" (dimensions of image, its max. resolution, etc.)
    - send client "base image" (coarsest resolution image)
      - optional data compression before injection into network
  - if (message := GET_MORE_DATA)
    - send client the image data for requested level in the foveal region around requested position
      - local data structures ensure same data is not retransmitted
      - optional data compression before injection into network
- **Image can be preloaded into memory or loaded on demand from disk**
  - we will assume for this project that it is preloaded

---

**COURSE PROJECT**

Designing an Adaptation Strategy for an Interactive Visualization Application

November 9, 1999
**Additional Details: Client**

Two threads

- **Processing thread**
  - send VIEW_NEW_IMAGE request to server
  - waits for response and saves received data into a structure
  - send one or more GET_MORE_DATA requests
  - correspond to new mouse position (or increments from previous one)
  - each request is composed of multiple messages, one per level
  - each message: (i, x, y, r)
  - wait for responses, one per level and save data into a structure

- **Display thread (simple message loop)**
  - set up display window and base image
  - wait for either a mouse click or additional display data
  - display image increments as and when they arrive

**Code Status**

- Application runs on Windows NT platforms

- TCP/Ethernet version of code will be available soon
  - will be installed by tomorrow on the machines of the PDSG NT lab (715 Broadway, 7th Floor)
  - can be downloaded tomorrow from http://www.cs.nyu.edu/vijayk/classproject.zip
  - send questions to fangzhe@cs.nyu.edu

- Remaining pieces will be available by next week
  - FM/Myrinet version of application
  - execution environment that permits control over application CPU share and network bandwidth
  - I shall describe both of these on Nov. 16th

**Adaptation in Active Visualization**

Metrics of interest

- Total download time
- Average Response Time
- Image resolution level

Parameters affecting behavior

- Image resolution level
- Fovea size/increment
- Choice of (if) compression algorithm

Resources affecting behavior

- CPU capacity
- Amount of physical memory
- Network bandwidth
- Disk bandwidth

**Application Resource Profiles**

[Graphs showing resource profiles]
Project Objective

Design an adaptation strategy that delivers the best performance for the active visualization application over a range of system load scenarios.

- Specifically, you need to modify the application so that it can detect changes in the following and appropriately adapt its behavior:
  - CPU load at the server
  - CPU load at the client
  - client-server bandwidth

- Adaptation can happen at:
  - the server
  - the client
  - the Myrinet network card

What Needs to Happen?

- **Server CPU load goes up**
  - offload work to network/client
    - go from a compute-intensive compression algorithm to no compression
    - okay to send the same data multiple times
  - degrade quality of connection
    - switch to a smaller frame size
    - switch to a lower level of resolution

- **Client CPU load goes up**
  - offload work to server/network
    - server assembles image for display
    - data is sent without compression
    - some processing happens on network coprocessor
  - degrade quality of connection

- **Network load goes up**
  - offload work to server/client
    - switch to a more compute-intensive/better compression algorithm
    - minimize amount of data that needs to be transmitted
  - degrade quality of connection

What does an Adaptation Strategy Involve?

- **How are changes in resource availability detected?**

- **When is adaptation triggered?**
  - e.g., what does low/high CPU load mean?

- **What happens as a result of the adaptation?**
  - i.e., which new behavior is chosen (from multiple choices)?
  - what are the policies guiding this selection (static/dynamic)?

- **How is adaptation implemented?**
  - e.g., what does it mean to switch compression algorithms when a message is in transit?

What Do I Expect?

- Two pieces
  - An adaptation strategy that (ideally) works for many applications
  - A realization of this strategy in the Active Visualization application
    - clean separation of different components of the strategy
    - okay to demonstrate only a subset of the overall strategy
      - e.g., adaptation in only the network coprocessor

What must be handed in

- Any code you develop
- A project writeup (no more than 20 pages)
- Due date: December 21, 1999
How to Get There?

1. Form project groups (2-3 people) today!
2. Become familiar with the code (10 days)
   - can use PDSG NT lab (715 Broadway, 7th Floor)
   - can also install on your own (the TCP/IP version)
3. Develop your own adaptation strategies
   - answer the questions on Slide 11
   - based on papers discussed in the class lectures
   - you can discuss this in class Nov. 16th and 23rd
     - each group will present every lecture
4. Implement your strategy and evaluate it
   - present updates in class on Nov. 30th and Dec. 7th

◆ Talk to me and Fangzhe