Distributed Visualization and Data Resources Enabling Remote Interaction and Analysis

Scott A. Friedman

UCLA Institute for Digital Research and Education

Motivating Challenge

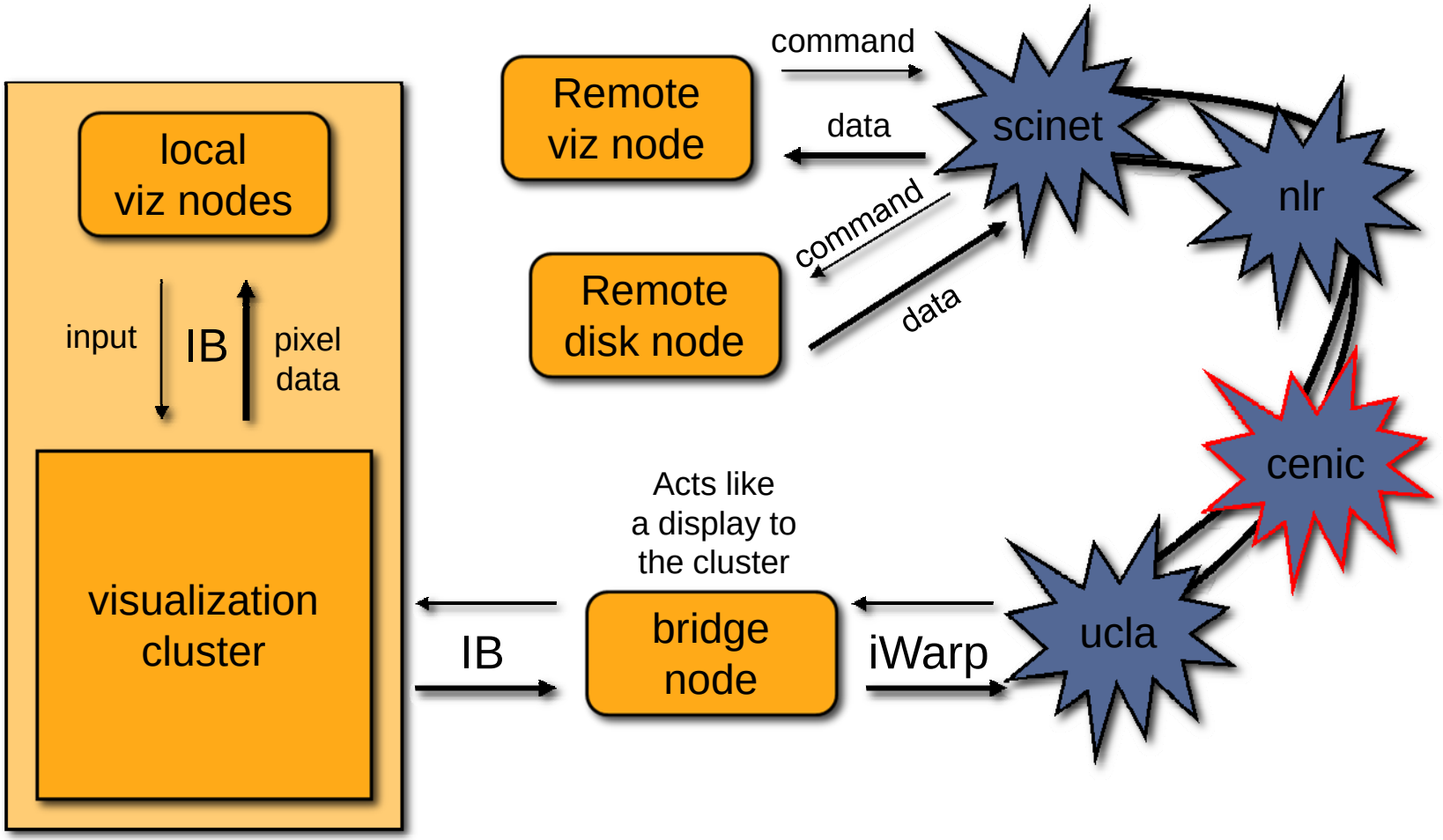
- Remote access to visualization cluster
  - Limited resources ($) for development
    - Re-use as much exiting IB (verb) code as possible (iWarp)
  - Leverage existing 10G infrastructure (UCLA, CENIC)
  - Use layer 3
  - Pacing critical for consistent user interaction
- Remote access to researcher's data
  - Same limitations as above
  - High performance retrieval of time step data
  - Synchronized to modulo of rendering rate

Visualization Cluster

- Hydra - Infiniband based
  - 24 rendering nodes (48 nvidia 9800GTX+)
    - 1 high-definition display node
    - 3 visualization center projection nodes
    - 1 remote visualization bridge node*
  - Research System - primarily
    - High performance interactive rendering (60Hz)
    - Dynamic (on-line) load balancing
  - System requires both low latency and high bandwidth
    - Perfect for Infiniband!

* We will return to this in a minute

Network Diagram



Remote Visualization Bridge Node

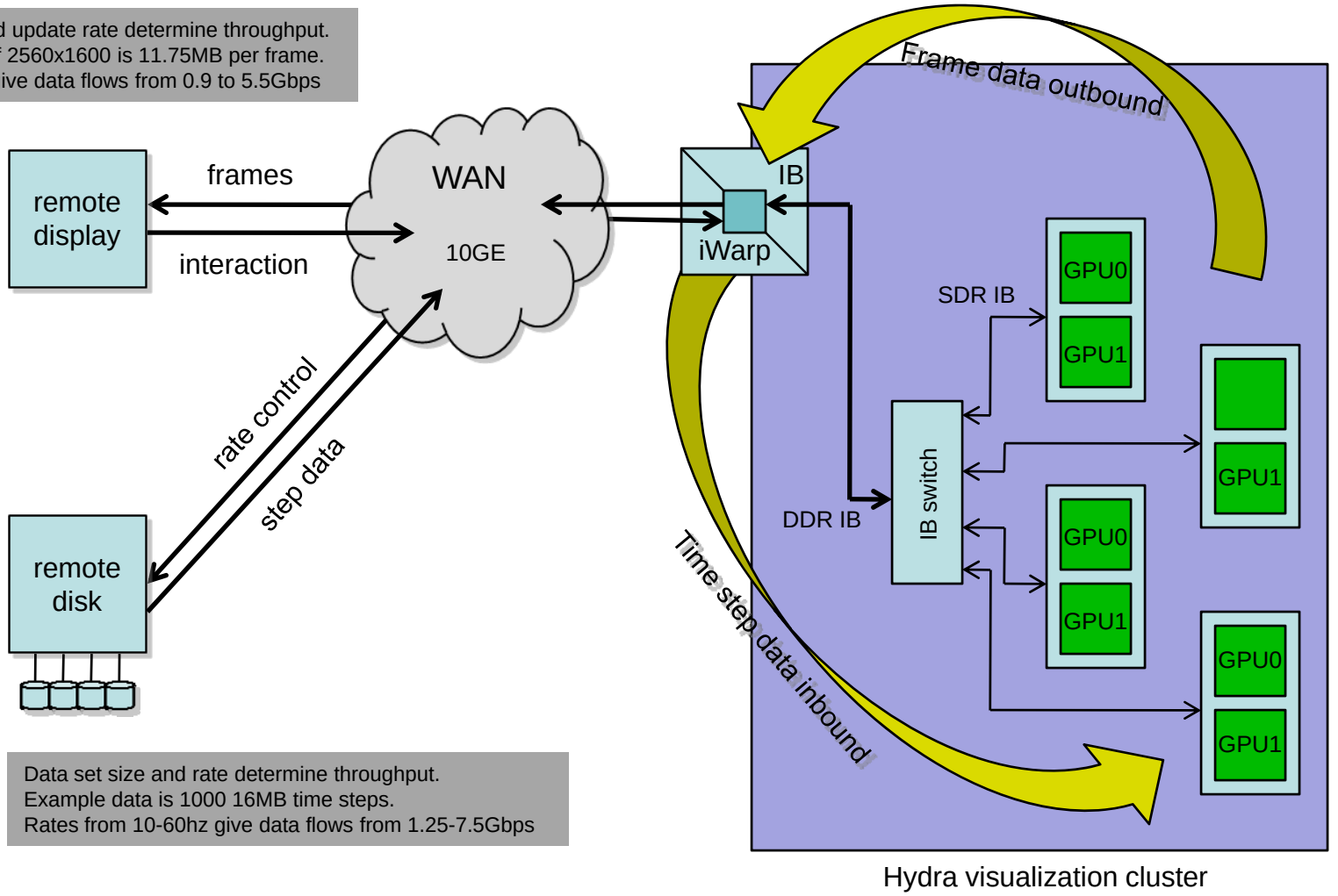
UCLA

OPENFABRICS ALLIANCE

- Bridges from IB to iWarp/10G Ethernet
- Appears like a display node to cluster
  - *No change to existing rendering system*
- Pixel data arrives over IB, which
  - is sent using iWarp to a remote display node
  - uses same RDMA protocol
    - Same buffer used for receive and send (bounce)
    - Very simple in principle - sort of…
    - TCP settings can be a challenge
- Time step data also passes through (opposite direction)
    - Distributed to appropriate rendering nodes

Data flow

Display resolution and update rate determine throughput.
Example resolution of 2560x1600 is 11.75MB per frame.
Rates from 10-60hz give data flows from 0.9 to 5.5Gbps



Data set size and rate determine throughput.
Example data is 1000 16MB time steps.
Rates from 10-60hz give data flows from 1.25-7.5Gbps

Hydra visualization cluster

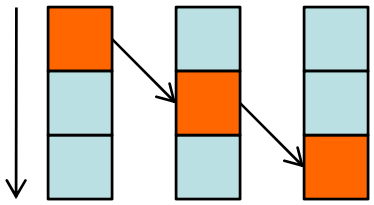
Example data set

- Uncompressed video stream
  - 20-60Hz at up to 2560x1600 ~ 1.8Gbps – 5.5Gbps
- Time steps
  - Time step size and rate determine flow
  - 10000 steps at 20-60Hz at 16MB/step ~ 2.4Gbps – 7.5Gbps
- Actual visualization
  - Interactive 100M particle n-body simulation
  - 11 unique processes
- Performance measurements are internal
  - Summaries are displayed at completion of run

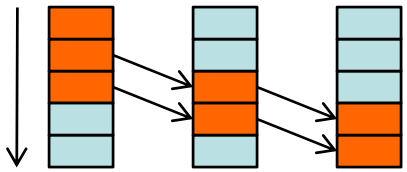
Running Demo

Future Work

- Latency is a non-issue at UCLA (which is our main focus)
  - ~60 usec across campus
- Longer latencies / distances *are* an issue
  - UCLA to UC Davis ~20ms rtt (CENIC)
  - UCLA to SC07 (Reno) ~14ms rtt (CENIC/NLR)
  - UCLA to SC08 (Austin) ~30ms rtt (CENIC/NLR)
- Frames and time step data completely in-flight (at high frame-rates)
  - Not presently compensating for this
- Minimizing latency important for interaction
  - Pipelining transfers helps hide latency: lat' = (lat / chunks) chunks + 1



chunks = 1 : lat = 2



chunks = 2 : lat = 1.5

Lessons Learned

- TCP nature of iWarp (some layer 3 related)
  - Congestion control wreaks havoc on interaction / latencies
  - Reliability
    - Not needed for video frames
    - Maybe needed for data (depends on application)
  - Buffer tuning – headache
- Layer 9 issues
- Better to use UDP?
  - Is there an unreliable iWarp?
- Dedicated/Dynamic circuits?
  - Obsidian Longbow IB extenders (haven't tried this)
  - Understand there may be an IP version out there? (not iWarp, but…)

Thank you

- Questions?
- friedman@ucla.edu