Application use of scheduled bandwidth
-the eaviv project-

Andrei Hutanu, Jinghua Ge, Cornelius Toole
Sandeep Nimmagadda, Gabrielle Allen
Goal

• Improve application performance by using distributed resources at various locations. Examples:
  • Powerful GPU cluster located at NCSA (Lincoln)
  • High-speed disk located at MU in Brno (Thor)
• Using networks can distribute applications that are not traditionally distributed
  • Interactive visualization!
• Improve capabilities by using unique resources not available locally

The eaviv Project
https://wiki.cct.lsu.edu/eaviv/

Andrei Hutanu (LSU) ahutanu@cct.lsu.edu
Approach

- Integrated application development
  - Remove layers, take into account all available resources
  - Strict separation in layers detrimental. Simplicity vs. performance trade-off
  - However, some applications could use access to intermediate layers!

- Networking example
  - Dominant design: TCP/IP
  - Information hidden
  - Cannot optimize. Easy to use
  - Want access to lower layers

The eaviv Project
https://wiki.cct.lsu.edu/eaviv/
Interactive visualization

• User(s) to visualize large (remote) datasets

• Use all available resources

The eaviv Project
https://wiki.cct.lsu.edu/eaviv/

Andrei Hutanu (LSU) ahutanu@cct.lsu.edu
The eaviv Project
https://wiki.cct.lsu.edu/eaviv/

Andrei Hutanu (LSU) ahutanu@cct.lsu.edu
How high-performance networks help visualization

- Increase I/O rate (recordings)
- Increase data size
  (top image: laptop only visualization, bottom image: distributed visualization on laptop using remote cluster)
- Collaborative visualization capabilities

The eaviv Project
https://wiki.cct.lsu.edu/eaviv/

Andrei Hutanu (LSU) ahutanu@cct.lsu.edu
Role of deterministic (scheduled) network

- Guarantee high-speed data transfer from the data server to the rendering cluster
  - Use remote storage that would otherwise need to be co-located with rendering. Can get faster network I/O rates compared to the local disk
- Full quality, minimum latency video streaming from rendering cluster to user
  - Use uncompressed video because it is possible; practically no data loss in the network
Demo (recorded)

- Parallel rendering on Lincoln (NCSA)
- Allocate ION network between NCSA and LSU
- Video stream to LSU 4k projector (about 500 Mbps bandwidth utilization)
- Interaction with visualization from LSU
- Can have multiple viewer + interaction points (various locations): Collaboration
- (data at NCSA on Lincoln)
Development/Execution Process

- Break application in components
  - Can execute at various locations
- set-up network connection to resources ..
- **Find available resources (rendering, data, network)**
  - At time of interest
- Select the optimal resources for application
- **Schedule and allocate resources**
- **Execute application**
Requirements / **Middleware** role

- If application does it all – very hard to write applications
- Deterministic co-allocation of resources
  - Rendering cluster (advance reservation used), storage
  - Network link (deterministic, manual “hope” co-allocation)
- Schedules exposed to applications/higher layers
  - Need to know in advance if resources will be available
- Network connecting to actual resources
  - Painful (at times) coordination process.
  - Campus issues
Critical current networking needs

- Multi-point circuit (or multiple point-to-point)
  - To connect more than two sites
  - Current options very limited and limiting: to connect NCSA, LSU and Brno 10G limit has to be split between two circuits that do not actually conflict with each other (unidirectional circuits may help)

- Debug packet loss
  - Takes a long time

- IP addressing (more technical) – but takes away from time that can be used to run apps
Other requirements (networking)

- Schedules exposed to app (what if questions)
  - To optimize application, need coordinated allocation of ALL resources
  - Need to know schedule in advance.
  - Prerequisite to co-allocation
- Topology visibility (information about network)
  - To optimize distributed application (frame rate, data rate, resolution) – specific to application
  - To decide what resources to use
- Phased commit (required for co-allocation)
NCSA
16 node ISL cluster (GPU + FPGA)

LSU
8 nodes rendering and 4 node data cluster

22TB

DCN through ICCN and OmniPop

55 Mpixel display, 10G WS

MU

CESNET

Other International Partners (G-Lambda, Phosphorus)

TeraGrid

DCN + SDN

SDN through ESnet

32 node CCS cluster

ORNL

8 nodes SPUR cluster

TACC

The eaviv Project
https://wiki.cct.lsu.edu/eaviv/

Andrei Hutanu (LSU) ahutanu@cct.lsu.edu

eaviv Grant NSF OCI #0947825