Interfacing CoUniverse & Internet2 DCN

Petr Holub, Miloš Liška, Andrew Lake, John Vollbrecht

Sitola – Laboratory of Advanced Networking Technologies
CESNET, Internet2, Masaryk University
Demo supported by: TAMU, StarLight

Internet2 Joint Tech Meeting
College Station, TX, 2009–02–02
Talk Overview

Motivation

CoUniverse

CoUniverse & Internet2 DCN

CoUniverse Operations

Implementation & Demo

Future Work
Motivation: CoUniverse

- High-definition collaborative environments
  - Using high-quality, high-definition media streams to build collaborative environment
    - bandwidth demands comparable to network link speeds (10 GbE) requires careful planning and configuration of infrastructure
    - lacks adaptivity to changing networking conditions
  - Large numbers of components needed to build the environment
    - each one of them needs to be configured
    - hard to orchestrate them manually to build the desired environment
    - virtually impossible to cope with network events manually
Motivation: CoUniverse

- CoUniverse is a framework to orchestrate interactive network-centric applications
  - GLIF and SuperComputing demonstrations
  - Introduction to High-Performance Computing class
- Fields where CoUniverse can help
  - audio/video transmissions
  - remote instrument control
  - generic network-centric application encapsulation
  - ... for scientific collaboration, telemedicine, etc.
Motivation: CoUniverse & Internet2 DCN

- Application-driven network allocation
  - if we can control applications, why not the network?
  - user should not be forced to allocate it manually
- CoUniverse is ideal for implementing this
  - it’s just another component to orchestrate
  - framework is general enough to implement it
CoUniverse Building Blocks

- **Organization of the CoUniverse**
  - Collaborative Universes where actual collaboration takes place
    - partitions virtual space (like, e.g., Virtual Venues)
    - provides privacy for users (may enforce authentication and authorization of users)
    - limits size of the system
  - Multiverse
    - registration and lookup of Collaborative Universes
    - automatically joined by each node

- **Network organization**
  - control plane based on P2P substrate
    - maximize robustness of the network
    - distributes control messages, updates from monitoring, etc.
  - one or more data planes running over native network
    - maximize performance for the applications (typically throughput, latency, jitter)
CoUniverse Building Blocks

- **Components**
  - nodes: physical nodes, having one or more network interfaces, running individual orchestrated applications
  - sites: aggregate of network nodes (e.g., you can tell you want stream from a specific site)
  - application groups: aggregate of applications

- **Application Group Controller (AGC)**
  - controls operations in the Universe
  - contains media streams scheduler if needed
  - one AGC per application group
Self-organization in CoUniverse

- Dynamic media streams scheduling:
  - Schedule media streams produced by media applications on particular network links (plan step)
  - scheduling media streams using bandwidth close to physical link capacity is hard
  - scheduling based on set of constraints
    - producer constraints, consumer constraints, data distribution constraints, network link constraints

- Resilience:
  - ability to react to changes/failures in the network infrastructure, media applications etc.
  - achieved by monitoring, infrastructure changes and/or failures lead to new media streams schedule
CoUniverse & Internet2 DCN – Demo

Network Topology

We want to meet over low-latency 1080i HD video using a network like this...

for switch addressing:
10.34.40.0/24
to be used on vlan3440
for debug and test
Motivation

CoUniverse

CoUniverse & Internet2 DCN

CoUniverse Operations

Implementation & Demo

Future Work

---

**CoUniverse & Internet2 DCN**

- DCN interfaces
  - web interface for humans
  - web service interface (w/ security!) for machines
- Initializing and tearing circuits
  - network link can have associated one or more lambda links
  - DCN-specific lambda link: two endpoints (including identification, IDC and (tagged|untagged) interface) and requested bandwidth
- Integration of on-demand circuits with CoUniverse brings another level of uncertainty into scheduling
  - should I count on a network link I’m uncertain to get?
  - should I preallocate the network? (but there are \( \frac{n^2-n}{2} \) bidirectional links!)
  - … see future work
CoUniverse Configuration

- **Node configuration**
  - *node information*: name, GPS coordinates for visualization
  - *interfaces*: IP addresses, identification of networks
  - *lambdas*: mapping of interfaces to available lambdas
  - *formats*: identification, bitrates, qualities
  - *applications/senders*: available formats
  - *applications/receivers*: available formats, requested source sites

- **AGC configuration**
  - stream scheduler (e.g., constraint-based scheduler, workflow support)
  - network control plugins (e.g., DCN)
CoUniverse Node Operations Diagram

- after joining a universe, an existing AGC is either joined or a new one is created
- sends topology to the AGC, including
  - end-to-end network topology
  - node configuration (site membership, interfaces, network connections, available applications)
  - requested sites for receiver applications
- waits for a new event from AGC (e.g., new plan) and monitors infrastructure to send updates to AGC
**MatchMaker**

- finds suitable source based on configuration of each receiver (if possible)
- builds plan based on available network features (links, reflectors)

**network initialization**

- added for end-to-end circuit initialization
- blocking stage to make sure we have the network prior to application startup
**Monitoring**

- **Media applications**
  - running or not after they were scheduled?

- **Data planes monitored in two modes**
  - more aggressive for network links actually used for scheduled media stream transmission
  - less aggressive for the links unused in any of data planes

- **Circuit monitor at AGC**
  - monitors changes of the allocated DCN circuit states
  - ensures renewing lambda allocation unless termination is requested
Visualisation in CoUniverse

- overview of actual CoUniverse state for the user
- network topology visualization
- actually scheduled media streams
- incorporation of network and application monitoring

GLIF 2007 visualisation

Current development visualisation
Notes on Implementation

- State of the CoUniverse
  - goal is to have usable proof-of-concept implementation
    - research on self-organization, application orchestration, scheduling
    - real-life applications (science, education)
  - https://www.sitola.cz/CoUniverse

- Open-source

- Implemented in Java, works on Linux, MacOS X, Windows

- P2P control plane
  - based on JXTA 2.4.1
  - joining takes 2–30 s depending on the network

- Scheduler
  - Choco 1.2.05 constraint solver
  - performance: $\approx 5$ s to compute and completely deploy a plan for reasonable Universe of 20 nodes (scattered "around the world")
Notes on Implementation

- Internet2 DCN
  - OSCARS API using Axis and Rampart
  - security based on X.509 certificates
- Internet2 DCN allocation performance
  - $\approx 3$ minutes to allocate the circuit from TAMU to StarLight
Demo

- CoUniverse manages
  - network: DCN circuits
  - application: UltraGrid
    - 1920×1080, 60 Hz, interlaced
    - DXT compressed video (decompression runs entirely in GPU)
    - 250 Mbps bandwidth
    - 130 ms latency end-to-end
    - Linux sender and receiver, over GE
Demo

Domain View

LEARN IDC

LEARN Domain

LEARN POP Houston, Texas

learn-sw c3750E

TAMU IDC

Texas A&M Domain College Station, Texas

tamu-sw c3750

tamu-sw c3750

c0Universe UltraGrid

10GE DataPlane
1GE DataPlane
ControlPlane

I2 IDC

Internet2 DCN Domain

I2 Ciena Chicago

I2 Ciena Houston

BRNO Czech Republic

hds.fi.muni.cz

Force10 E300 Prague

CESNET Czech Republic

Starlight E1200

Starlight Chicago, Illinois
Future Work

- **CoUniverse**
  - ongoing research into scheduling
    - constraint based, graph based, genetic algorithms, user-friendly workflow support
  - improving code quality
  - direct support for more applications

- **DCN**
  - security needs to be scalable and developer-friendly
  - improvements in allocation speed of the DCN circuits
    - important for interactive applications (“I want the circuit NOW!”)
  - multi-point issues
  - renewing/modification of reservations
  - notification API instead of status polling
Future Work: Multi-Point Connections

- The more components we have, the more options we have:

```
\begin{align*}
&\text{\(rfl_1\)} \quad \text{\(s_1\)} \quad \text{\(rcv_1\)} \\
&\text{\(s_1\)} \quad \text{\(rfl_1\)} \quad \text{\(rcv_1\)} \quad \text{\(rcv_2\)} \quad \text{\(rcv_3\)} \quad \text{\(rcv_4\)} \\
&\text{\(s_2\)} \quad \text{\(rfl_2\)} \quad \text{\(rcv_1\)} \quad \text{\(rcv_2\)} \quad \text{\(rcv_3\)} \quad \text{\(rcv_4\)}
\end{align*}
```
Future Work: Possible Ways

- **Multi-point API**
  1. I need to interconnect the following nodes \( (x_1, \ldots, x_n) \)
  2. (optional) I have the following constraints (e.g., I need capacity between \( x_i \) and \( x_j \))
     - …this starts to be very tricky when there are multiple paths
  3. Tell me topology of what you have constructed

- **Alternatively – available topology API**
  - tell me topology of links available at this moment
  - simulating this by individual link polling results in \( O(n^2) \) requests
Acknowledgments

- Texas A&M University team:
  Bob Arnold, Sergio Chacon
  J. D. Lewis, Steve Wiederhold, The Texas A&M Networking Group
- ISI East: Tom Lehman
- Internet2: Brian Cashman
- Masaryk University: Tomáš Rebok
Thank you for your attention!

Q?/A!

<hopet@ics.muni.cz>