Lancaster University

Experiments with GTS at Lancaster University: A Retrospective

20th October 2015

Matthew Broadbent

Disclaimer

The views and opinions expressed in this presentation are those of the authors and do not necessarily reflect the hard work, effort and cost that went into building these facilities! No offence is intended ☺ This is a users point of view.

Lancaster University

- Lancaster started SDN experimentation back in 2012
- Participants in the 2nd Open Call of the OFELIA Project
- ‘OpenFlow-assisted VoD’
- Investigated the usage of OpenFlow as a tool to redirect requests for (video) content
- Built a prototype content delivery platform to evaluate the suitability of the technology
- Demonstrated potential efficiency gains
- Spawned a number of papers and demonstrations

Lancaster University

- Simple 4 node topology: one video client, one cache node, one origin server and a controller (network and cache) – each a virtual machine
- Spanned three testbeds (i2CAT, iMinds and ETH Zürich)
- iMinds acted as a transit with no nodes
- Provisioned using the OFELIA Control Framework
- Time-consuming process only possible through a GUI (no programmatic control)
- Required manual authorisation from each island
- Prevented rapid experimentation
- Limited in the resources that could be used (4 islands federated)

Lancaster University

- Gave us scope and scale beyond what was possible in our small lab
- This really lends gravity to evaluation and helps with paper acceptance
- Process is not without it flaws
- Building and operating testbeds is **hard**, takes **time** and costs **money**
- The tools needed to be built to support this work: none existed

Lancaster University

- Lancaster joined the 2nd Open Call of the Fed4FIRE project
- 'Multi-testbed Experimentation of a Video-on-Demand Distribution Service' – *MEVDDS*
- Continuing our content distribution work
- Using different resources within the facility to expand our experiments
- Extended the prototype to include a configurable API
- Allows cache behaviour to be defined on-the-fly
- Built and evaluated two examples applications to demonstrate the suitability of the interface: *load balancer & failover monitor*

Lancaster University

- Different model to resource allocation
- Provisioning through a specification
- Based on the the GENI RSpec format
- Mismatch in tools across different testbeds: *Omni, jFed etc.*
- Made the process a significant bottleneck
- Spent more time setting up the experiment than developing and actually executing!
- Resources, compatibility and availability continuously changing

GÉANT

Lancaster University

- Lancaster joined the GÉANT OpenFlow Facility - *GOFF*
- 'Cross-site Evaluation of an OpenFlow-assisted Video-on-Demand Distribution Service' – *CEOVDS*
- Scaling up our previous experimentation on OFELIA
- Spanned 5 physical locations
- Multiple clients, multiple cache nodes
- Stress-tested prototype in a realistic scenario: content delivery networks are naturally distributed across multiple locations
- Took our experimentation to a new level of scale

Lancaster University

- GUI-based resource allocation, similar to OFELIA
- Used the same toolsets, modified slightly
- No need to wait for reservation authorisation
- Public-facing interface on virtual machines made access simpler
- **Stability!** Production rather than research
- Advanced notice of downtime and upgrades
- Allowed us to concentrate on experimentation
- Produced some great results published in a number of papers/journals
- Interesting to compare hardware/software switch performance between GOFF and OFELIA – these choices matter!

Looking Forward

Lancaster University

- Our work is progressing more towards the use of flexible and virtualised infrastructures
- How can we provision a content delivery network on demand?
- How do we dynamically allocate resources in the most appropriate locations, pre-populated with relevant content?
- How do we discover and provision the network and its resources? What are the interfaces to do so?

Continued Development

Lancaster University

- After the conclusion of these projects, we moved experimentation ‘in-house’
- Development of prototype continues:
  - OpenFlow v1.3 support: multi-table functionality
  - OpenStack integration: create, deploy, modify and destroy cache nodes as required
  - Further extensions to the API to allow even richer cache-centric applications to be built
  - Support for new/different delivery techniques: HTTP/2, multicast
- Initial design work; how do we move it back out to a testbed?
- Is there a middle-ground? Integration between local and large-scale development and evaluation

What do we want from a testbed?

Lancaster University

- Continue to offer the resources required to give research significant scale and real-world applicability
- Flexible yet powerful experimental definitions, particularly programmatic ones
- Common toolchain without exceptions or peculiarities
- Stability and support so that experimenters can concentrate on experimentation
  - Double-edged sword: ultimately still research networks that need upgrading and improving
- The ability to interface with the testbed from *within* the experiment
  - Saves us from having to nest virtualisation

Planned Experimentation

Lancaster University

- We are planning on using the GTS testbed for the next phase of evaluation; vCDN use case
- Even more scale (transatlantic connectivity!)
- Hardware *and* software switches
- More interesting /realistic dataplane behaviour: beyond a L2 learning switch
- The ability to provision new nodes and resources from within our own control framework and experiment (to be tested!)
- Interesting to see how the GTS interface compares to existing APIs, such as OpenStack

Lancaster University

Questions?

Matthew Broadbent – m.broadbent@lancaster.ac.uk