The SAVI Testbed for Software-Defined Infrastructure

CCECE 2014:
Workshop on Networking and Cloud Computing Test Beds

Presented by: Thomas Lin
t.lin@mail.utoronto.ca
Overview

• SAVI and the SAVI Testbed
• Software-Defined Infrastructure… Why?
• The SAVI Approach to SDI
• FPGA Virtualization on SAVI
• Current Status and Projects
• Ongoing Work and Conclusion
SAVI

• What is SAVI?
  – Smart Applications on Virtual Infrastructure
  – Canadian partnership between industry, academia, research, and education networks

• Research aim:
  – Explore designs regarding Future Internet
  – Address designs of future applications platforms built on virtualized infrastructure
• **Multi-Tiered Cloud**
  – Core nodes
    • Massive-scale, many resources
  – Smart Edge nodes
    • Smaller scale, may be connected to access points
SAVI Testbed: Core Nodes

• Core nodes: Massive scale
  – Distant from end-users
  – Placed at locations with inexpensive or renewable energy
SAVI Testbed: “Smart Edge” Nodes

Wireless/Optical Access Points

“Smart Edge” Datacentres

BackBone Network

Massive-Scale Core Datacentres

• Smart Edge nodes: Smaller scale
  – Located closer to end-users
  – For services requiring low-latency, local processing
  – More variety in types of resources
    • e.g. GPUs, FPGAs, miniBEEs, SDRs, etc.
Current SAVI Testbed Deployment

Diagram showing the connections and deployment of SAVI testbeds at various locations including UVic, McGill, Calgary, Waterloo, Carleton, YorkU, with ORION at the center and SAVI TB Control Center.
Software-Defined Infrastructure? Why?!
IaaS at a Glance

• Infrastructure as a Service (IaaS)
  – Provides “infrastructure resources”
    • e.g. virtual machines, storage, network firewalls, etc.
  – Resources may be self-provisioned and self-managed via a set of open interfaces (i.e. APIs)
IaaS Management

• Traditional cloud management
  – Primarily concerned with the compute resources
  – Limited network management offered in “blocks”
    • e.g. Pool of IPs, firewall rules, load balancers, etc.

• Finer-grain network management may be achieved *separately* via SDN principles
  – SDN controller directly configures network switches via an open protocol (e.g. OpenFlow)
IaaS Management cont.

• Is a cloud manager and a separate SDN controller enough…
  – to address end-to-end QoS?
  – to fulfill contracts and other SLA’s?

• Application performance influenced by both underlying compute and network resources
  – e.g. Streaming video service
Software-Defined *Infrastructure* (SDI)

- SDI aims to integrate the management and control of *all infrastructure components*…
  - Compute + Networking + Storage + Access

- While providing open interfaces for users…
  - e.g. RESTful APIs for enabling self-service

- All within a centralized control plane
  - Control logic implemented in software
The SAVI Approach to SDI
SDI RMS

- SDI Resource Management System (RMS)
SAVI SDI RMS

- **SDI Manager**
  - “Command centre” for integrated control and management

- **Topology Manager**
  - Database of resources, their specifications, relationships, and real-time status
SAVI SDI RMS

- Proxy resource controllers:
  - OpenStack for managing the cloud
  - SDN controller for managing the network
- Controls and manages both virtual & physical
SAVI SDI Manager

- A “pluggable” controller, comprised of:
  - A module manager
  - One or more control and management modules

- Modules may implement custom RESTful APIs (open interfaces) for:
  - Querying infrastructure information
  - Self-managing and self-provisioning of resources
  - Providing configuration and status updates
SAVI Topology Manager

- Stores configuration and specs of all resources
- Monitors resource state via resource controllers
  - e.g. OpenStack for cloud, OpenFlow for network
- Stores information in a graph database (Neo4j)
  - Graph DB enables mapping the topology of the testbed
SDI: Potential Modules

- Information provided by OpenStack, SDN controllers, and the topology manager opens the door for:
  - Green networking
  - Fault tolerance
  - Traffic engineering
  - Application QoS enforcement
  - Network-aware VM migration
  - Real-time monitoring and diagnostics
Heterogeneous Resources

• Currently provided by SAVI Testbed:
  – FPGAs: NetFPGAs, BEE2s, miniBEEs, DE5-Nets
  – GPUs, SDRs

• Integration of new resources into the testbed
  – Added to Nova in OpenStack as new flavours
FPGA Virtualization on SAVI

Work done by Stuart Byma

Presented at TRIDENTCOM 2014
Resource Virtualization

• SAVI aims to fully virtualize the infrastructure

• Smart Edge contains non-conventional resources
  – e.g. GPUs, FPGAs, SDRs, etc.
    • Goal is to have these virtualized as well
    • Ongoing investigation and research

• First step towards FPGA virtualization has been realized in the SAVI testbed
FIELD-PROGRAMMABLE GATE ARRAYS
FIELD-PROGRAMMABLE GATE ARRAYS

© H. Roesner

© flickr / MadPhysicist
Hardwired FPGA functions

- ARM A9
- ARM A9
- SRAM
- SRAM
- SER Des
- DSP
- 1GE MAC

May 4th, 2014
Example: Smith-Waterman DNA sequencing (Dynamic Programming)

49x – 980x speedup (I/O dependent) on Xilinx V4-LX160 FPGA vs. 2.2GHz AMD Opteron

(Storaasli/Cray 2009)
SAVI Testbed FPGA Resources

• BEE2 Systems
  – Four FPGAs, lots of RAM, 10Gb Ethernet
  – In one box!
SAVI Testbed FPGA Resources

- FPGA Cards
  - NetFPGAs (10 GE and 1 GE) and Altera DE5-Nets
  - Mounted inside physical servers
  - On board RAM
  - PCIe to Motherboard
Applications

- Pure computational acceleration over PCIe

- 10G Ethernet ports enable…
  - Custom NICs, offload software processing
  - Line rate packet processing for
    - Custom protocols
    - Future internet router prototypes
Taking it Further

• FPGA cards in baremetal servers
  – Relatively non-virtualized
  – One physical resource, not sharable
    • Not as flexible/scalable
    • Wasteful if the entire chip is not utilized

• We want **Fully Virtualized Hardware**
FPGA Partial Reconfiguration (PR)

- Fix? Use PR!
- Virtualized FPGA Resources (VFR)
- Multi-tenant
- Static logic contains chip level IO, memory controllers, etc.
  - Allows admins to police VFR interfaces (security)
Now we can “boot” a network connected FPGA accelerator on demand, in seconds!
Example: VFR Load Balancer

- Important task in large scale systems
- UDP load balancer
  - Hardware analyzes payload of packet and load balances based on parsed data
VM vs VFR -- Throughput

- Load balancer throughput comparison
- Same logic implemented in both
  - Virtual Machine (VM)
  - Virtualized FPGA Resource (VFR)

<table>
<thead>
<tr>
<th>VM</th>
<th>VFR</th>
</tr>
</thead>
<tbody>
<tr>
<td>25 MB/s</td>
<td>&gt; 100 MB/s</td>
</tr>
</tbody>
</table>
Where’s SAVI now?

Where is it going?
Current Status

- SDI with deployed in all SAVI datacentres
  - All resources connected via 1 and 10 GE links
  - Running modules on SDI:
    - Scheduling module
    - Network Control module

- Supported resources include:
  - Virtual machines, Baremetal servers, GPUs, various FPGAs, ATOM servers, network fabric

- Preliminary experiments /w access points
  - Enables inclusion of mobile devices into testbed experiments
Currently Running Projects

• Reconfigurable Hardware Virtualization (FPGA)

• Big Data Analysis
  – Apache Hadoop & Spark clusters

• Content Centric Networking (CCN)
  – X-CCN: Extended CCN to support long term subscription

• Connected Vehicle and Smart Transportation (CVST)

• Wireless integration
  – Enabling integration of mobile devices into the SAVI testbed

• … and more!
Ongoing Testbed Work

• Monitoring and Measurement
  – Measuring, collecting, analyzing, and reporting resource state information

• Resource Scheduling
  – Intelligent resource allocation and automatic scaling

• Traffic Engineering and Path Selection
  – Dynamic datacentre traffic engineering using SDN

• Green Networking

• Virtualization of other non-conventional resources and access points
Conclusion

• SAVI aims to enable rapid deployment, maintenance and retirement of applications
• SDI enables integrated management of cloud, network, and other types of resources
  – Deployed in Canadian national SAVI testbed
  – An infrastructure-aware controller & manager
• Increasing heterogeneity of infrastructure resources requires a virtualized approach
  – Goal: Enable shared access to all resources for concurrent experimentation by various researchers
Thank You!

Questions & Answers
Addendums
Software-Defined Infrastructure
Detailed SAVI Architecture
Computing with FPGAs

• Fully customized dataflow and buffering

• Tightly coupled pipelining of computations

• Very low energy / computation ratio

• Computation cores can be switched out in msecs with partial reconfiguration
FPGA programmability drawbacks

• Implementation (compile-equivalent) takes **hours**
• Established Hardware Description Languages (Verilog HDL, VHDL) are very low-level
• Downside of design flexibility: No established programming models
• Interfaces are often specific to vendors, platforms, FPGA boards
User Designed Hardware

- Standard stream interface to VFRs
- Template Verilog file
- Script-based compile and image packaging