TRANS CLOUD: Design Considerations for a High-Performance Cloud Architecture Across Multiple Administrative Domains

Rick McGeer, HP Labs

For the TransCloud Team: HP Labs, UC San Diego, University of Victoria, Northwestern University, University of Amsterdam, TU-Kaiserslautern, Princeton University, PlanetWorks, PlanetLab, GENI, G-Lab, DFN, NLR, GLIF

August 1, 2010
• TransCloud: A Cloud Where Services Migrate, Anytime, Anywhere In a World Where Distance Is Eliminated
  – Joint Project Between GENICloud, iGENI, G-Lab
  – GENICloud Provides Seamless Interoperation of Cloud Resources Across N-Sites, N-Administrative Domains
  – iGENI Optimizes Private Networks of Intelligent Devices
  – G-Lab contributes networking and advanced cloud resources
“The Cloud” offers the prospect of ubiquitous information and services…BUT…

- Performance of Cloud services Highly Dependent On Location
  - Of End-User, Applications, Middle Processes, Network Topology
  - Of Cloud Data, Compute Processes, Storage, etc

Why?
- Performance of Legacy Protocols Highly Dependent on Latency

Therefore:
- If the Clouds Are Too Far Away, Performance Will Be Very Severely Restricted

Ergo
- Clouds Needs To Be Close To Service Sites OR
- Networks (And Clouds) Must Be Designed To Eliminate Distance
Context 2: Living With Legacy Protocols Over Commodity Internet vs Creating Alternatives

- Legacy Is There For a Reason
  - Compatibility
  - Fairness
  - Congestion Avoidance

- Therefore: Distributed Cloud
  - Minimal Latencies Over Legacy Internet To Anywhere/Everywhere

- Therefore: Private Internal Networks
  - Eliminate Latency Dependence Internally
  - Use Aggressive Internal Transport/Application Protocols
    - TIA-1039, Reliable Blast UDP, Lambda RAM
    - Flow Control Enabled
Context 3: No Cloud Lives Everywhere

- Clusters are much easier to build than points-of-presence
- Most commercial clouds today have only a few sites
- Therefore: cloud service providers want to run services across *multiple* clouds
  - Need a cloud standard that offers identical interfaces over multiple domains
- Inspiration: the web
  - Standard protocol for sending documents
  - Standard document format
  - Permission and access control on a site-by-site, page-by-page basis
Context 4: General Considerations

- Major Cloud Use Case: Big Data, Distributed Collection, Must Live With Available Networks
  - Smart Cities
  - Sensor Nets
- Best Case: Create Private Network
  - Owning Optical Fiber
  - Create High Performance Wireless Point-to-Point Links
- Many Data Intensive Science Projects, Including
  - High Energy Physics (e.g. LHCNet, Science Data Network, I-WIRE)
  - Atmospheric Sensing Apparatus
  - Ocean Observing (e.g., Project Neptune)
  - Distributed Radio and Optical Telescopes
  - Telemedicine
Premise: Compute Where Data Lives!

- Computation is Ubiquitous and Easy To Obtain
- Programs Are Small and Easy to Transmit
- Most Programs \textit{Reduce} Data
- Often Data Is Large and Challenging To Transmit
  - E.g., Jim Gray distributing SDSS by sending computers by FedEx!
- \textbf{Solution} -- Send Programs to Data
- Requires
  - High-performance, low-latency network
  - Common API’s and operating environments
  - \textit{Lightweight, user-based federation}
What do we need to make this work?

• Advanced Networking and Caching
  – Firm guarantees on bandwidth and latency on a per-application basis
  – Application support at Layer 3 and Layer 2
  – *Means: Private Network where possible*

• Access to platforms wherever data lives
  – *But data lives everywhere!*
  – No organization has Points of Presence (PoP)s everywhere
  – Need for an individual to be able to make arrangements with a cloud service provider, anywhere, efficiently, minimal overhead
  – *Common form of identity*
  – *Common identity not required*
  – *Common AUP not required*
What do we need to make this work?

• Ability to instantiate and run a program anywhere
  – Common API at each level of the stack
  – IaaS/NaaS (VM/VN Creation)
  – PaaS (guaranteed OS/Programming environment)
  – OaaS (Standard Query/Data Management API)

• Easy, Standard Naming Scheme
  – I need to know the name of my VM’s, logins, store etc without asking
Solution – TransCloud

• Introducing TransCloud Prototype
  – An Early Instantiation of the Architecture
  – A Distributed Environment That Enables Component and Interoperability Evaluation
  – A Testbed On Which Early Experimental Research Can Be Conducted
  – An Environment That Can Be Used To Explain/Showcase New Innovative Architecture/Concepts Through Demonstrations
TransCloud Today

Approx 40 nodes at 4 sites, 10 Gb/s connectivity
• Sites at
  – HP Labs, Palo Alto
  – UC San Diego
  – Northwestern
  – Kaiserslautern

• Tomorrow (*literally!*)
  – Amsterdam

• Connectivity provided by:
  – CAVEWave, StarLight, NetherLight, DFN, National Lambda Rail, Global Lambda Integrated Facility
Demo

(code by Chris Pearson and Chris Matthews, University of Victoria, data store from Paul Muller (Kaiserslautern) and Michael Zink(U Mass))
• Multi-site query example
  – Internet data repository (packet traces)
    • Kaiserslautern, Germany (thanks to Paul Muller)
    • UC San Diego (thanks to Michael Zink)
  – Run an analysis job at each site
  – Transmit the results back to HP Labs
  – Run summary job at HPL
• What’s being demonstrated?
  – Ability to run *multi-site* job
  – Sending programs to data
  – Prototype of analysis of coming world of sensors
<table>
<thead>
<tr>
<th>Institution</th>
<th>Open Jobs</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCSD</td>
<td>No Open Jobs</td>
</tr>
<tr>
<td>HP</td>
<td>No Open Jobs</td>
</tr>
<tr>
<td>Kaiserslautern</td>
<td>No Open Jobs</td>
</tr>
<tr>
<td>Northwestern</td>
<td>No Open Jobs</td>
</tr>
<tr>
<td>Amsterdam</td>
<td>No Open Jobs</td>
</tr>
</tbody>
</table>

**Transcoding Statistics**
Several Basic TransCloud Concepts

- High Performance Highly Distributed Cloud Architecture Allowing Processes Across Multiple Administrative Domains Integrated With Dynamic Networking (GENI)
- Scalable Lightweight Federation Processes
- Services Are Based On Processes That Can Be Executed Anywhere World-Wide (Location Independent)
- Top Level Services Can Be Accessed Via Public Internet
- Core Processes and Data Streams Leverage Sophisticated Communication Services Not Merely “Best Effort” Commodity Internet
TransCloud Distributed Query Demo

<table>
<thead>
<tr>
<th>Location</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>UVic</td>
<td>No Open Jobs</td>
</tr>
<tr>
<td>UCSD</td>
<td>Status of Open Jobs at site UCSD</td>
</tr>
<tr>
<td>HP</td>
<td>No Open Jobs</td>
</tr>
<tr>
<td>Kaiserslautern</td>
<td>Status of Open Jobs at site Kaiserslautern</td>
</tr>
<tr>
<td>Northwestern</td>
<td>No Open Jobs</td>
</tr>
<tr>
<td>Amsterdam</td>
<td>No Open Jobs</td>
</tr>
</tbody>
</table>

Transcoding Statistics
Introduction – TransCloud

- TransCloud Architectural Components
  - High Level APIs
  - A High Performance General Programming Environment
  - High Levels of Virtualization Based on VMs and Network Abstractions
TransCloud Equals

- IaaS Based on Slice-Based Federation Architecture (GENI/FIRE Standard)
  - Current instantiation: MyPLC over Eucalyptus
  - Want: ports to OpenStack, etc.
- Identity: X.509 certificates and ssh keys
  - TransCloud sites agree to accept these as forms of identity
  - *Which* to accept up to the site
- Standard DNS Infrastructure
  - `<instanceName>..<sliceName>..<siteName>..<authorityName>..trans-cloud.net`: experiment interface
    - e.g. `hadoop22.queryTest.hplabs.genicloud.trans-cloud.net`
  - `<siteName>..<authorityName>..trans-cloud.org`: admin interface
    - `hplabs.genicloud.trans-cloud.org`
  - Each authority does its own DNS.
• Experimental QaaS (Distributed Hadoop/Pig)
• User-done PaaS (some stock images, but the usual tools for building your own…)

TransCloud Equals..
Integration with GENI

- Programmer and User Interface to Cluster Control is MyPLC
  - Cluster version of PlanetLab control interface
  - Used for a number of clusters worldwide, including VICI project in US

- Mechanics of cluster control done by Eucalyptus
  - Single Eucalyptus user – MyPLC
  - Users log in to MyPLC, issue directives, MyPLC effectuates by issuing appropriate Eucalyptus commands
### TransCloud Architecture

<table>
<thead>
<tr>
<th>Distributed Pig</th>
<th>Distributed Hadoop</th>
<th>NaClRePy</th>
<th>GENI Eucalyptus</th>
<th>1039/RBUDP…</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Slice Federation</td>
<td>Flow Primitives</td>
</tr>
<tr>
<td>Architecture</td>
<td></td>
<td></td>
<td>Architecture</td>
<td></td>
</tr>
</tbody>
</table>
TransCloud Distributed Query
On April 15 (about) we were attacked by the Romanian Black Hats
- Stock VM had a privileged user with a guessable password
- Came with the VM…
- Attack was a worm attack to recruit bots for botnets
- We were alerted when a third-party site saw worm probes coming from us

Solution: shut it down, fix it, bring it up

The Fix:
- Use MyPLC (PlanetLab) as the controller
- Login only by ssh key, X.509 cert (GENI standard)
- Ssh login only from specified IP addresses (EC-2 standard)
- Authorized users can add whitelisted IP’s
- Currently enforced by iptables, but we’ll add support into OpenFlow

Running final pre re-launch tests now
Goals for 2011

- Complete integration with MyPLC
- Integrate the ProtoGENI Resource Specification (RSpec)
  - Modified to make sense for clusters
- Integrate the GENI standard Authorization-Based Access Control (ABAC)
- Add utility to permit users to manually adjust connectivity rules
  - Integration with ProtoGENI RSpec
Advancing TransCloud

- If You Are Interested In Using This Environment, Contact Us

- If You Would Like To Contribute Resources, Contact Us
• THANKS!

• Questions????