BUILDING AND RUNNING OPENSTACK ON POWER8

LANCE ALBERTSON

Oregon State University Open Source Lab

@ramereth
SUMMARY

- POWER8 Overview
- POWER at OSUOSL
- Building a RHEL-based P8 platform with Openstack
- Architecture porting issues
- Problems we have encountered with OpenStack
- Openstack deployment with Chef
- OSL Wrapper cookbook
- Next Steps
DISCLAIMER
POWER8 OVERVIEW

DESIGN

- Designed to be a massively multithreaded chip
- Designs are available for licensing under the OpenPOWER Foundation
- Little-Endian & Big-Endian
- Several non-IBM companies building P8 hardware
  - Tyan, Rackspace (OpenCompute-based) & Google
OPENPOWER ABSTRACTION LAYER (OPAL)

- OPAL is the new Open Source firmware for POWER8
- Acts as an on-system HMC
- Enables the machine to boot similar to PC servers
- Linux Kernel and loads the boot loader Petitboot
- Petitboot provides a shell environment for debugging and setup
- Petitboot will use kexec and boot into the system kernel
HISTORY

- Providing PPC64 compute resources since 2005
- Close collaboration with IBM LTC
- POWER5, POWER7 and now POWER8
- OSL managed LPAR deployment to make it easier on projects
- Pre-P8 Projects:
  - Debian, Gentoo, Fedora, PostgreSQL
  - Linux Foundation, Haskell, GoLang
  - Mozilla, OpenSUSE, LLVM, GCC
POWER8 AT OSUOSL

- Goal is to provide on-demand PPC64/PPC64LE compute resources to FOSS projects
- Assist with ppc64/ppc64le porting & testing
- Expose OSU students to OpenStack and POWER8
- Collaboration with IBM engineers on architecture issues
- Create a vanilla Openstack cluster for FOSS projects
PROJECTS RUNNING ON OUR P8 CLUSTER

- CloudFoundry, Docker, CentOS, CouchDB
- Haskell, Glibc, JXcore, LLVM, NodeJS
- OpenJDK, GoLang, oVirt, libjpeg-turbo
- BLCR, Gentoo
SUPPORTED OS PLATFORMS

PowerKVM
Ubuntu
RHEL
DECISION TO USE RHEL

- Little community support at the time and opportunity to help the community
- We use CentOS internally as our primary OS & more familiar with the RHEL eco-system
- RHEL has the RDO OpenStack distribution that is well supported
- Chef support with OpenStack needed some help
- I love challenges!
OPENSTACK ARCHITECTURE (OLD)

- Started in 2014
- Icehouse
- Controller node
  - Runs all public API services, dashboard
  - DB hosted on a shared bare-metal system
  - X86_64 CentOS 6 VM running on Ganeti+KVM
- Compute node(s)
  - Nova compute and networking
  - Flat networking
  - PPC64 Fedora
OPENSTACK ARCHITECTURE (NEW)

- Deployed 2016 (deployed last week)
- Mitaka
- Controller node
  - Runs all public API services, dashboard
  - DB hosted on a shared bare-metal system
  - X86_64 CentOS 7 VM running on Ganeti+KVM
- Compute node(s)
  - Nova Compute
  - Neutron Networking
    - Linuxbridge
    - Provider and Tenant networking using VXLAN
  - PPC64LE CentOS 7.2
COMPUTE NODES

- Did initial development on Fedora 19
- Fedora 20 PPC64 base system (old)
- Fedora 21 versions of a few packages
- CentOS 7.2 PPC64LE base system (new)
ARCHITECTURE PORTING ISSUES

CHEF

- No PPC64/PPC64LE Chef client
- Needed to build our own chef-client
- Omnibus
  - Bootstrap build env
  - Build dependency issues
  - Architecture configuration issues in Omnibus
- Chef has stable ppc64/pc64le builds today
PACKAGE SUPPORT

• Support for P8 was bleeding edge and new features were added weekly
• Built versions of latest packages from Fedora rawhide packages:
  ▪ qemu
  ▪ libvirt
  ▪ kernel
• Internal repo for these custom packages:
  ▪ http://ftp.osuosl.org/pub/osl/repos/yum/openpower/centos-7/ppc64le/
• Kernel required a few custom options to be enabled
• Runtime setup: Disable SMT
GUEST OS IMAGES

- Few OS supported ppc64/ppc64le or provided guest images pre-built
- Variety of tools which are platform specific
- Missing support for cloud-init
- Initially started creating images manually with qemu directly
PACKER -- MULTI PLATFORM SUPPORT

- We needed Go to use Packer
- GoLang support was literally in the works
- Finally built our own packer binary last Nov!
- WIP Packer Templates:
  - [https://github.com/osuosl/bento/tree/ramereth/ppc64](https://github.com/osuosl/bento/tree/ramereth/ppc64)
ARCHITECTURE ISSUES

- OPAL firmware bugs
- pre-P8 machines were very buggy
- IPMI console would sometimes stop working
- Random lockups
- Included HW RAID, but no cached write-back support
LEARNING AND UNDERSTANDING OPENSTACK

- Lots of moving pieces
- Neutron networking is complex and a moving target
- Deciding on the proper design architecture for our use case
BUGS AND "FEATURES"

- Interaction between libvirt and nova-compute was buggy at times
- Some bugs were just Icehouse itself, others were architecture specific
- Learning how to deploy Openstack and making (gasp) mistakes!
- Iptables issues between Chef and Openstack
- Provider networks configures dnsmasq as an open resolver
- SSL API endpoints
STABILITY

- Rabbitmq would constantly need to be restarted
- nova-compute services would randomly stop working
- Running Fedora on compute and CentOS on controller made things ... interesting
RHEL / CENTOS SUPPORT

- Introduced in 7.1 and fully supported in 7.2
- CentOS community was still bootstrapping and testing
- We built our own pre-release CentOS 7.2 for testing
- Using ppc64le on compute nodes
RDO

- Community for deploying Openstack on CentOS, Fedora and RHEL
- Repositories built against each Platform
- Each release of OpenStack separated
RHEV (RED HAT ENTERPRISE VIRTUALIZATION)

- Updated KVM packaging
- Part of the Virt SIG of CentOS
- Used SRPMs to build ppc64le versions in a location repo
- One patch needed to work around bug
WHY CHEF?

- Primary CM tool used at the OSL
- Provides a lot of testing capability on deployment
- Can use the full power of the Ruby language for configuring the cluster
CHEF OPENSTACK

- Set of cookbooks that will deploy the various services of Openstack
- Part of the OpenStack umbrella
- Community driven
- Did a major refactoring of the code for Mitaka release
OSL OPENSTACK

- Created a wrapper cookbook (osl-openstack)
- [https://github.com/osuosl-cookbooks/osl-openstack](https://github.com/osuosl-cookbooks/osl-openstack)
- OSL site specific configuration
- Split recipes out by upstream cookbook name
- Contains ppc64le specific changes
- Currently only tested on CentOS 7
RECIPES/DEFAULT.RB

- recipes/default.rb
- Default configuration for cluster
- Include local yum repos
- Include command clients
- Logic around endpoints
RECIPES/IDENTITY.RB

- recipes/identity.rb
- Just includes recipes
- Some wrapper, some upstream
- Allows us to test just Keystone by itself
Recipes/Controller.rb

- recipes/controller.rb
- Pulls in all wrapper recipes needed to build a controller
- Allows for us to split things out eventually if we want to
TESTING AND DEVELOPMENT

- Unit Testing
  - ChefSpec
  - RSpec
- Integration Testing
  - Test Kitchen
  - ServerSpec
- Chef Provisioning
  - Deploy VMs as controller/compute
  - Deploy on bare-metal for a test cluster
UNIT TESTING

- Ensure the Chef code is doing what it’s supposed to do
- Easily test Architecture-specific logic
- Verify configuration files contain proper settings
- Examples:
  - spec/default_spec.rb
  - spec/compute_controller.rb
  - spec/linuxbridge_spec.rb
UNIT TESTING (OUTPUT)

$ rspec spec/default_spec.rb

osl-openstack::default
  includes cookbook base::ifconfig
  includes cookbook selinux::permissive
  includes cookbook yum-qemu-ev
  includes cookbook openstack-common
  includes cookbook openstack-common::logging
  includes cookbook openstack-common::sysctl
  includes cookbook openstack-identity::openrc
  includes cookbook openstack-common::client
  includes cookbook openstack-telemetry::client
setting arch to x86_64
  does not add OSL-Openpower repository on x86_64
setting arch to ppc64
  add OSL-openpower-openstack repository on ppc64
TEST KITCHEN & SERVERSPEC

• Test Kitchen
  ▪ Test CLI tool which allows you to execute the configured code on one or more platforms
  ▪ Integrates with testing frameworks
  ▪ Must have tool for Chef users
  ▪ Configured via `.kitchen.yml`

• ServerSpec
  ▪ RSpec tests for configured servers
  ▪ Integration tests
  ▪ Ensures things actually happen on the system
  ▪ Example: `test/integration/default/serverspec/default_spec.rb`
$ kitchen list

<table>
<thead>
<tr>
<th>Instance</th>
<th>Driver</th>
<th>Provisioner</th>
<th>Verifier</th>
<th>Transport</th>
<th>Last Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>default-centos-72</td>
<td>Openstack</td>
<td>ChefZero</td>
<td>Busser</td>
<td>Rsync</td>
<td>Not Created</td>
</tr>
<tr>
<td>mon-centos-72</td>
<td>Openstack</td>
<td>ChefSolo</td>
<td>Busser</td>
<td>Rsync</td>
<td>Not Created</td>
</tr>
<tr>
<td>mon-controller-centos-72</td>
<td>Openstack</td>
<td>ChefSolo</td>
<td>Busser</td>
<td>Rsync</td>
<td>Not Created</td>
</tr>
<tr>
<td>ops-messaging-centos-72</td>
<td>Openstack</td>
<td>ChefZero</td>
<td>Busser</td>
<td>Rsync</td>
<td>Not Created</td>
</tr>
<tr>
<td>identity-centos-72</td>
<td>Openstack</td>
<td>ChefZero</td>
<td>Busser</td>
<td>Rsync</td>
<td>Not Created</td>
</tr>
<tr>
<td>image-centos-72</td>
<td>Openstack</td>
<td>ChefZero</td>
<td>Busser</td>
<td>Rsync</td>
<td>Not Created</td>
</tr>
<tr>
<td>network-centos-72</td>
<td>Openstack</td>
<td>ChefZero</td>
<td>Busser</td>
<td>Rsync</td>
<td>Not Created</td>
</tr>
<tr>
<td>linuxbridge-centos-72</td>
<td>Openstack</td>
<td>ChefZero</td>
<td>Busser</td>
<td>Rsync</td>
<td>Not Created</td>
</tr>
<tr>
<td>compute-controller-centos-72</td>
<td>Openstack</td>
<td>ChefZero</td>
<td>Busser</td>
<td>Rsync</td>
<td>Not Created</td>
</tr>
<tr>
<td>compute-centos-72</td>
<td>Openstack</td>
<td>ChefZero</td>
<td>Busser</td>
<td>Rsync</td>
<td>Not Created</td>
</tr>
<tr>
<td>dashboard-centos-72</td>
<td>Openstack</td>
<td>ChefZero</td>
<td>Busser</td>
<td>Rsync</td>
<td>Not Created</td>
</tr>
<tr>
<td>block-storage-centos-72</td>
<td>Openstack</td>
<td>ChefZero</td>
<td>Busser</td>
<td>Rsync</td>
<td>Not Created</td>
</tr>
<tr>
<td>block-storage-controller-centos-72</td>
<td>Openstack</td>
<td>ChefZero</td>
<td>Busser</td>
<td>Rsync</td>
<td>Not Created</td>
</tr>
<tr>
<td>telemetry-centos-72</td>
<td>Openstack</td>
<td>ChefZero</td>
<td>Busser</td>
<td>Rsync</td>
<td>Not Created</td>
</tr>
</tbody>
</table>
$ kitchen test default

---> Starting Kitchen (v1.8.0)
---> Cleaning up any prior instances of <default-centos-72>
---> Destroying <default-centos-72>...
    Finished destroying <default-centos-72> (0m0.00s).
---> Testing <default-centos-72>
---> Creating <default-centos-72>...
    OpenStack instance with ID of <a25fa410-5caf-4f96-bddb-1e6dadd06d9> is ready.
...

Chef Client finished, 115/198 resources updated in 03 minutes 12 seconds
Finished converging <default-centos-72> (3m41.31s).
---> Setting up <default-centos-72>...
Finished setting up <default-centos-72> (0m0.00s).
---> Verifying <default-centos-72>...
Preparing files for transfer
INFRASTRUCTURE NEXT STEPS

• Add Nagios checks (DONE!)
• Continue to fix bugs and other issues as they come up
• Rebuild old Icehouse cluster as Mitaka (no upgrade)
• Add support for object storage
• Update documentation
• Add support for non-live migration
• Mellanox networking
PROJECT EXPERIENCE

- Improve and streamline on boarding process
- Expand cluster’s disk storage capacity
- Improve stability of the cluster
- Add more projects!
- Submit your request:
  - [http://osuosl.org/services/powerdev/request_hosting](http://osuosl.org/services/powerdev/request_hosting)
QUESTIONS?

Lance Albertson
lance@osuosl.org
@ramereth


Links:

- http://github.com/ramereth/presentation-openstack-power8
- https://github.com/osuosl-cookbooks/osl-openstack
- http://osuosl.org/services/powerdev/request_hosting

Attribution-ShareAlike CC BY-SA ©2016