End of the Road - Facing Current Scaling Limits within OpenStack

OpenStack Summit, Vancouver

May 19, 2015

Christian Berendt
Cloud Solution Architect
B1 Systems GmbH
berendt@b1-systems.de

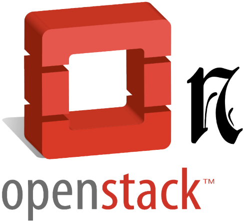
Thomas Kaergel
Linux Consultant & Developer
B1 Systems GmbH
kaergel@b1-systems.de

Introducing B1 Systems

- founded in 2004
- operating both nationally and internationally
- more than 60 employees; low employee turnover
- Provider for IBM, SUSE, Oracle & HP
- vendor-independent (hardware and software)
- Focus:
  - Consulting
  - Support
  - Development
  - Training
  - Operations
  - Solutions

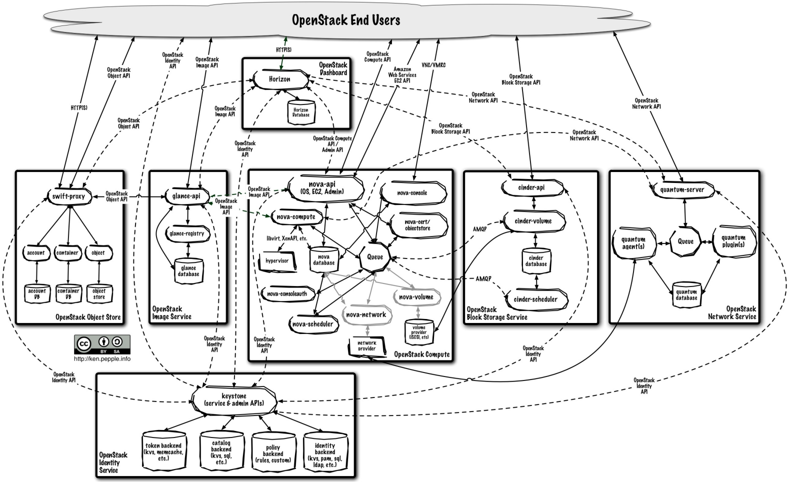
Areas of Expertise

- Virtualization (XEN, KVM & RHEV)
- Systems management (Spacewalk, Red Hat Satellite, SUSE Manager)
- Configuration management (Puppet & Chef)
- Monitoring (Nagios & Icinga)
- IaaS Cloud (OpenStack & SUSE Cloud)
- High availability (Pacemaker)
- Shared Storage (GPFS, OCFS2, DRBD & CEPH)
- File Sharing (ownCloud)
- Packaging (Open Build Service)
- Providing on-site systems administration and/or development



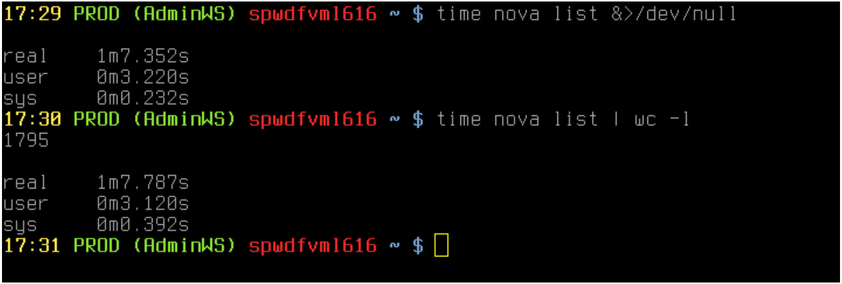
nce upon a time ...

Source: lassedesignen/Shutterstock.com



Source: Ken Pepple, http://ken.pepple.info/

Source: varunsingh180000/Pixabay.com



```
17:29 PROD (AdminWS) spwdfvml616 ~ $ time nova list &>/dev/null

real	1m7.352s
user	0m3.220s
sys	0m0.232s
17:30 PROD (AdminWS) spwdfvml616 ~ $ time nova list | wc -l
1795

real	1m7.787s
user	0m3.120s
sys	0m0.392s
17:31 PROD (AdminWS) spwdfvml616 ~ $ 
```

What Happenend?

Observations

- nova list extremely slow
- almost all nova operations on instances affected
- horizon too slow to be usable
- DB and nova services under heavy load

Observations

- nova list extremely slow
- almost all nova operations on instances affected
- horizon too slow to be usable
- DB and nova services under heavy load

Observations

- nova list extremely slow
- almost all nova operations on instances affected
- horizon too slow to be usable
- DB and nova services under heavy load

Observations

- nova list extremely slow
- almost all nova operations on instances affected
- horizon too slow to be usable
- DB and nova services under heavy load

Case Study

Many Instances in Single Tenant (Folsom)

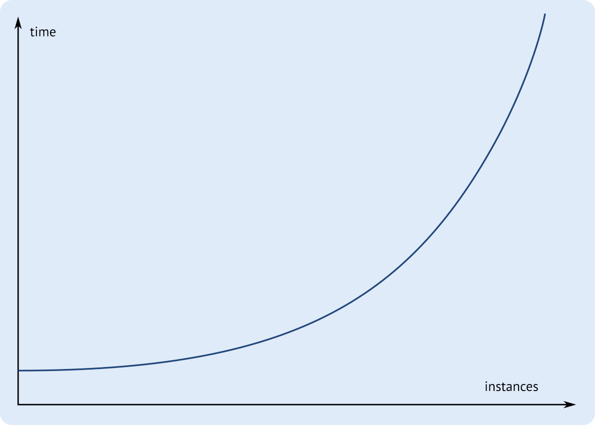


Figure : nova-list Duration over Instance Count

Many Instances in Multiple Tenants (Folsom)

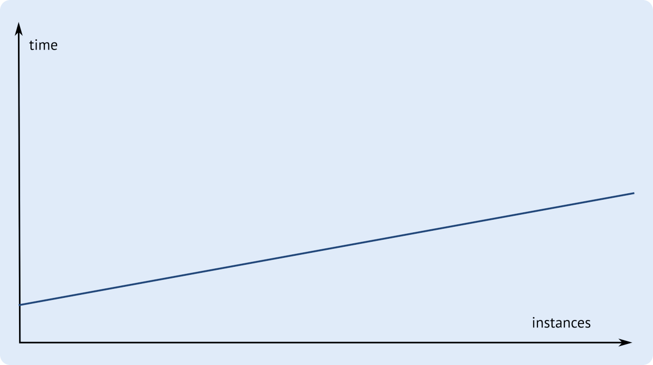


Figure : nova-list Duration over Instance Count

Many Instances in Single Tenant (Today)

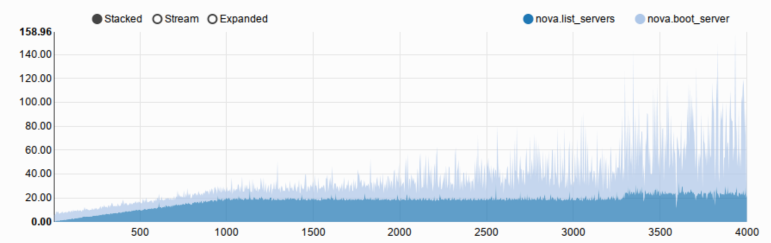


Figure : nova-list Duration over Instance Count

Investigation Strategy

Source: OpenClips/Pixabay.com

Actions

- watch CPU load on infrastructure during load situation
- switch logmode to debug
- observe logs during load situation
- turn mysql query logging on and watch the DB queries
- analyze the code

Actions

- watch CPU load on infrastructure during load situation
- switch logmode to debug
- observe logs during load situation
- turn mysql query logging on and watch the DB queries
- analyze the code

Actions

- watch CPU load on infrastructure during load situation
- switch logmode to debug
- observe logs during load situation
- turn mysql query logging on and watch the DB queries
- analyze the code

Actions

- watch CPU load on infrastructure during load situation
- switch logmode to debug
- observe logs during load situation
- turn mysql query logging on and watch the DB queries
- analyze the code

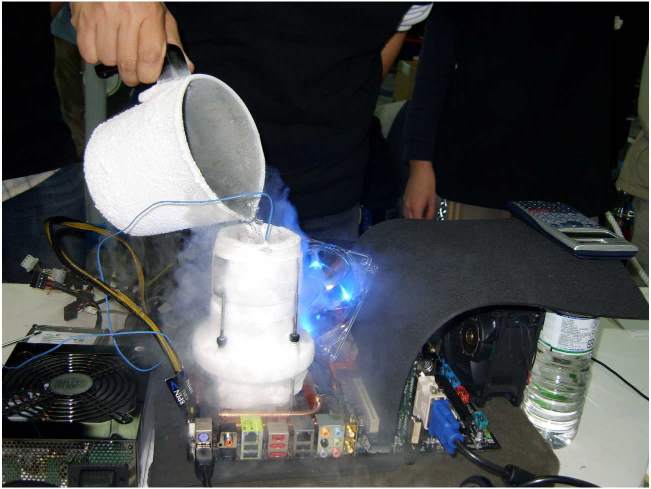
Actions

- watch CPU load on infrastructure during load situation
- switch logmode to debug
- observe logs during load situation
- turn mysql query logging on and watch the DB queries
- analyze the code

sqlalchemy DB-Join, linewidth 80 characters, length 100+ lines

```
SELECT instances.created_at AS instances_created_at,
instances.updated_at AS instances_updated_at,
instances.deleted_at AS instances_deleted_at,
instances.id AS instances_id,
instances.user_id AS instances_user_id,
instances.project_id AS instances_project_id,
instances.image_ref AS instances_image_ref,
instances.kernel_id AS instances_kernel_id,
[...]
...
...
[...]
WHERE instances.deleted = 0 AND instances.host = 'computexen0158'
```

Possible Solutions



Source: RicoShen/Wikimedia.org

- more powerful hardware for Nova and DB
- rewrite nova/sqlalchemy code that generates those big DB joins
- reorganize user/tenant-layout for the use case

- more powerful hardware for Nova and DB
- rewrite nova/sqlalchemy code that generates those big DB joins
- reorganize user/tenant-layout for the use case

- more powerful hardware for Nova and DB
- rewrite nova/sqlalchemy code that generates those big DB joins
- reorganize user/tenant-layout for the use case

Prevention Strategy

- determine expected load and expected elasticity
- design for horizontal scalability using active/active HA setups whenever possible
- built a representative miniature of your cloud for measurements, experiments and development

Useful Tools

Vagrant

Source: Fco.plj/de.wikipedia.org

Vagrant Advantages

- reproducible and portable work environments
- easy to set up and learn
- usable for scale testing and development
- many *providers* available (Virtualbox, KVM, VMware...) to virtualize hosts
- choice between many *provisioners* (Shell, Ansible, Chef, Puppet...) to configure hosts

Vagrant Advantages

- reproducible and portable work environments
- easy to set up and learn
- usable for scale testing and development
- many *providers* available (Virtualbox, KVM, VMware...) to virtualize hosts
- choice between many *provisioners* (Shell, Ansible, Chef, Puppet...) to configure hosts

Vagrant Advantages

- reproducible and portable work environments
- easy to set up and learn
- usable for scale testing and development
- many *providers* available (Virtualbox, KVM, VMware...) to virtualize hosts
- choice between many *provisioners* (Shell, Ansible, Chef, Puppet...) to configure hosts

Vagrant Advantages

- reproducible and portable work environments
- easy to set up and learn
- usable for scale testing and development
- many *providers* available (Virtualbox, KVM, VMware...) to virtualize hosts
- choice between many *provisioners* (Shell, Ansible, Chef, Puppet...) to configure hosts

Vagrant Advantages

- reproducible and portable work environments
- easy to set up and learn
- usable for scale testing and development
- many *providers* available (Virtualbox, KVM, VMware...) to virtualize hosts
- choice between many *provisioners* (Shell, Ansible, Chef, Puppet...) to configure hosts

Example Vagrant Environment

- hardware with >8 cores and >32 GB RAM
- capable of hosting all OpenStack controller hosts full-scale
- Vagrant provider Virtualbox
- most-used provisioners: Shell, Ansible and Puppet

Example Vagrant Environment

- hardware with >8 cores and >32 GB RAM
- capable of hosting all OpenStack controller hosts full-scale
- Vagrant provider Virtualbox
- most-used provisioners: Shell, Ansible and Puppet

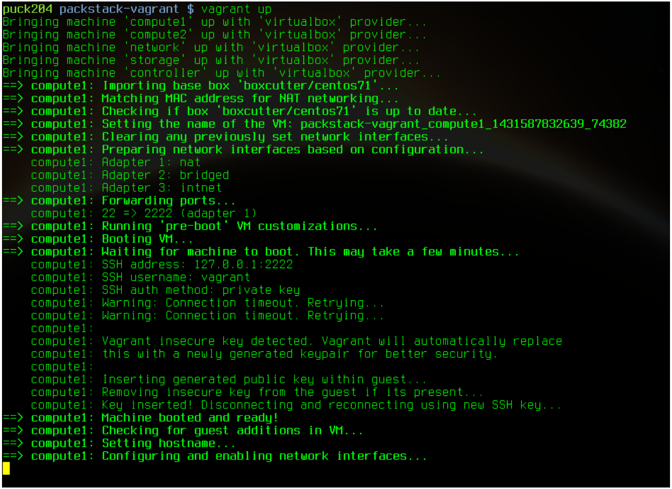
Example Vagrant Environment

- hardware with >8 cores and >32 GB RAM
- capable of hosting all OpenStack controller hosts full-scale
- Vagrant provider Virtualbox
- most-used provisioners: Shell, Ansible and Puppet

Example Vagrant Environment

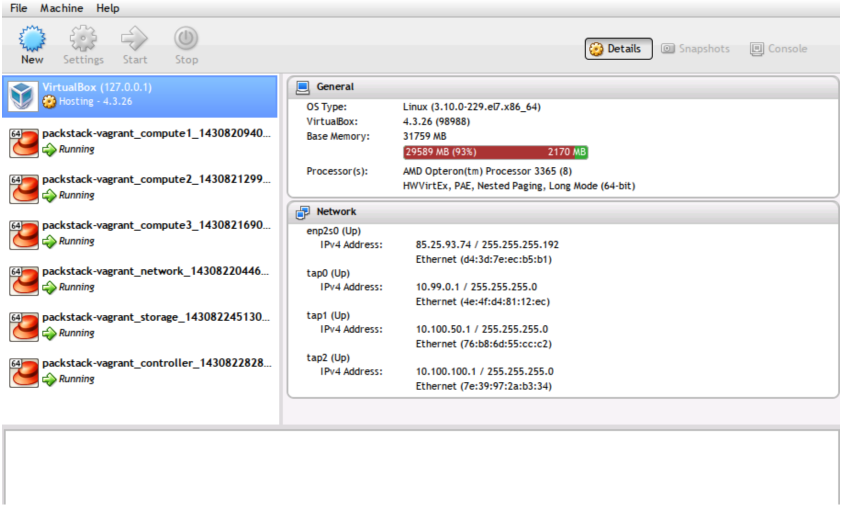
- hardware with >8 cores and >32 GB RAM
- capable of hosting all OpenStack controller hosts full-scale
- Vagrant provider Virtualbox
- most-used provisioners: Shell, Ansible and Puppet

Vagrant at Work...



```
puck204 packstack-vagrant $ vagrant up
Bringing machine 'compute1' up with 'virtualbox' provider...
Bringing machine 'compute2' up with 'virtualbox' provider...
Bringing machine 'network' up with 'virtualbox' provider...
Bringing machine 'storage' up with 'virtualbox' provider...
Bringing machine 'controller' up with 'virtualbox' provider...
==> compute1: Importing base box 'boxcutter/centos71'...
==> compute1: Matching MAC address for NAT networking...
==> compute1: Checking if box 'boxcutter/centos71' is up to date...
==> compute1: Setting the name of the VM: packstack-vagrant_compute1_1431587832639_74382
==> compute1: Clearing any previously set network interfaces...
==> compute1: Preparing network interfaces based on configuration...
    compute1: Adapter 1: nat
    compute1: Adapter 2: bridged
    compute1: Adapter 3: intnet
==> compute1: Forwarding ports...
    compute1: 22 => 2222 (adapter 1)
==> compute1: Running 'pre-boot' VM customizations...
==> compute1: Booting VM...
==> compute1: Waiting for machine to boot. This may take a few minutes...
    compute1: SSH address: 127.0.0.1:2222
    compute1: SSH username: vagrant
    compute1: SSH auth method: private key
    compute1: Warning: Connection timeout. Retrying...
    compute1: Warning: Connection timeout. Retrying...
    compute1:
    compute1: Vagrant insecure key detected. Vagrant will automatically replace
    compute1: this with a newly generated keypair for better security.
    compute1:
    compute1: Inserting generated public key within guest...
    compute1: Removing insecure key from the guest if its present...
    compute1: Key inserted! Disconnecting and reconnecting using new SSH key...
==> compute1: Machine booted and ready!
==> compute1: Checking for guest additions in VM...
==> compute1: Setting hostname...
==> compute1: Configuring and enabling network interfaces...
```

Running Environment



VirtualBox (127.0.0.1)
Hosting - 4.3.26

packstack-vagrant_compute1_1430820940...
Running

packstack-vagrant_compute2_1430821299...
Running

packstack-vagrant_compute3_1430821690...
Running

packstack-vagrant_network_14308220446...
Running

packstack-vagrant_storage_143082245130...
Running

packstack-vagrant_controller_1430822828...
Running

General

OS Type: Linux (3.10.0-229.el7.x86_64)
VirtualBox: 4.3.26 (98988)
Base Memory: 31759 MB
29589 MB (93%) 2170 MB
Processor(s): AMD Opteron(tm) Processor 3365 (8)
HWVirtEx, PAE, Nested Paging, Long Mode (64-bit)

Network

enp2s0 (Up)
IPv4 Address: 85.25.93.74 / 255.255.255.192
Ethernet (d4:3d:7e:ec:b5:b1)

tap0 (Up)
IPv4 Address: 10.99.0.1 / 255.255.255.0
Ethernet (4e:4f:d4:81:12:ec)

tap1 (Up)
IPv4 Address: 10.100.50.1 / 255.255.255.0
Ethernet (76:b8:6d:55:cc:c2)

tap2 (Up)
IPv4 Address: 10.100.100.1 / 255.255.255.0
Ethernet (7e:39:97:2a:b3:34)

OpenStack Rally

Source: wpaphotomotorsport/Pixabay.com

OpenStack Rally Advantages

- easy usage and setup
- many benchmark templates available which already cover many standard situations
- Rally plugins enable easy creation of more complex and use-case-specific benchmarks
- nice presentation of results

OpenStack Rally Advantages

- easy usage and setup
- many benchmark templates available which already cover many standard situations
- Rally plugins enable easy creation of more complex and use-case-specific benchmarks
- nice presentation of results

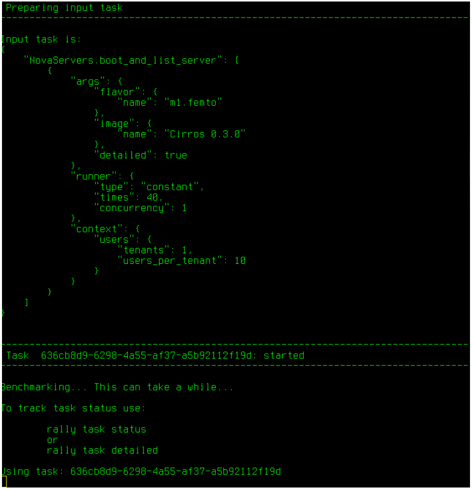
OpenStack Rally Advantages

- easy usage and setup
- many benchmark templates available which already cover many standard situations
- Rally plugins enable easy creation of more complex and use-case-specific benchmarks
- nice presentation of results

OpenStack Rally Advantages

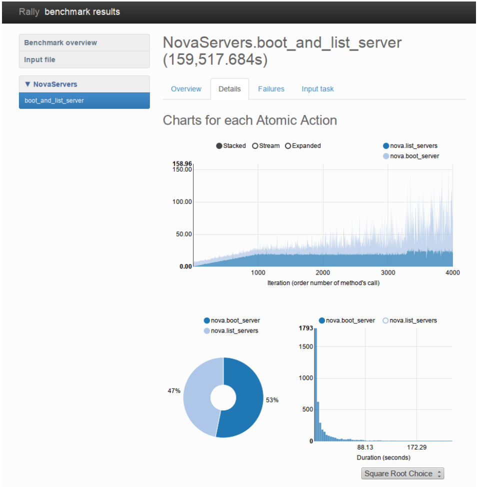
- easy usage and setup
- many benchmark templates available which already cover many standard situations
- Rally plugins enable easy creation of more complex and use-case-specific benchmarks
- nice presentation of results

OpenStack Rally in Action...



```
 Preparing input task
--------------------------------------------------------------------------------

Input task is:
{
    "NovaServers.boot_and_list_server": [
        {
            "args": {
                "flavor": {
                    "name": "m1.femto"
                },
                "image": {
                    "name": "Cirros 0.3.0"
                },
                "detailed": true
            },
            "runner": {
                "type": "constant",
                "times": 40,
                "concurrency": 1
            },
            "context": {
                "users": {
                    "tenants": 1,
                    "users_per_tenant": 10
                }
            }
        }
    ]
}


--------------------------------------------------------------------------------
 Task  636cb8d9-6298-4a55-af37-a5b92112f19d: started
--------------------------------------------------------------------------------

Benchmarking... This can take a while...

To track task status use:

        rally task status
        or
        rally task detailed

Using task: 636cb8d9-6298-4a55-af37-a5b92112f19d
```

OpenStack Rally



Rally benchmark results

NovaServers.boot_and_list_server (159,517.684s)

Charts for each Atomic Action

Fake Drivers

- simulate instances or volumes
- transparent for the OpenStack controller hosts
- independent from hardware requirements

Fake Drivers

- simulate instances or volumes
- transparent for the OpenStack controller hosts
- independent from hardware requirements

Fake Drivers

- simulate instances or volumes
- transparent for the OpenStack controller hosts
- independent from hardware requirements

Nova Fake Driver Configuration

- Fake Nova Compute Driver

nova.conf

```
...
# Driver to use for controlling virtualization. Options
# include: libvirt.LibvirtDriver, xenapi.XenAPIDriver,
# fake.FakeDriver, baremetal.BareMetalDriver,
# vmwareapi.VMwareVCDriver, hyperv.HyperVDriver (string value)
compute_driver=fake.FakeDriver
...
```

Conclusion

- determine clear design specifications (max instances, volumes, elasticity, users, tenants)
- use Rally to thoroughly test your setup within the specs
- perform a full-scale test without FakeDrivers prior to go-live
- use active/active HA setups for the core services to retain horizontal scalability

- determine clear design specifications (max instances, volumes, elasticity, users, tenants)
- use Rally to thoroughly test your setup within the specs
- perform a full-scale test without FakeDrivers prior to go-live
- use active/active HA setups for the core services to retain horizontal scalability

- determine clear design specifications (max instances, volumes, elasticity, users, tenants)
- use Rally to thoroughly test your setup within the specs
- perform a full-scale test without FakeDrivers prior to go-live
- use active/active HA setups for the core services to retain horizontal scalability

- determine clear design specifications (max instances, volumes, elasticity, users, tenants)
- use Rally to thoroughly test your setup within the specs
- perform a full-scale test without FakeDrivers prior to go-live
- use active/active HA setups for the core services to retain horizontal scalability

Thank You!

For more information, refer to info@b1-systems.de
or +49 (0)8457 - 931096