Can we further boost HPC Performance?
Integrate IBM Power System to OpenStack Environment
(Part 1)
Ankit Purohit, Research Engineer
NTT Communications
1. Our Background
2. Providing GPU Resources: P100
3. Benchmarking with different tool: DGX-1 vs S822LC for HPC(Minsky)
4. I/O advantages of IBM Power System
5. Performance improvement with Memory Interleave: nbody
6. Summary
Agenda

1. Our Background
2. Providing GPU Resources: P100
3. Benchmarking with different tool: DGX-1 vs S822LC for HPC (Minsky)
4. I/O advantages of IBM Power System
5. Performance improvement with Memory Interleave: nbody
6. Summary
Background

- NTT Communications is the largest Telecommunications company in Japan and has subsidiaries and offices in over 110 cities worldwide.
- Part of a Fortune Global 100 company.
- NTTCom offers Artificial Intelligence communication engine like COTOHA.
- Currently, we provide GPU(M60,P100 and V100 [x86]) cloud using OpenStack.
Our current situation

- We provide instances to users using following GPU cards with various flavors on x86 servers.
We need more computing for “Deep Learning”
How about IBM POWER8?

- POWER8 has more power than x86
- The difference between POWER8 and DGX-1 (x86) will be discussed soon in part 2.

IBM Power System S822LC (Minsky)
Agenda

1. Our Background
2. Providing GPU Resources: P100
3. Benchmarking with different tool: DGX-1 vs S822LC for HPC (Minsky)
4. I/O advantages of IBM Power System
5. Performance improvement with Memory Interleave: nbody
6. Summary
Providing GPU Resources with P100 GPU

- We provide instances to in-house users with and without GPU on our private cloud environment with following.

OpenStack using pci Passthrough with KVM
Private Cloud with OpenStack

1. Nova-api receives GPU-VM request from user

   Nova-api receives GPU-VM request from user

   Nova Controller Node

   Nova Compute Node

   Nova-scheduler

   Rabbitmq

2. Nova-scheduler determines which compute node to allocate

   Nova-scheduler determines which compute node to allocate

   3. Nova-compute launches GPU-VM using Libvirt with KVM PCI-passthrough on nova compute node

   Nova Instance: d60.v4
   (Without GPU)

   Nova Instance: g2-TeslaP100

   Nova-api

   Nova-compute

   Libvirt

   QEMU/KVM

   QEMU/KVM

   Launch Instance

   Flavor of DGX-1

   Flavor Details

   Nova-api

   Nova-compute

   Nova-scheduler

   Nova Controller Node

   Nova Compute Node

   Fig. Nova architecture
Our aim

- OpenStack uses KVM
- Can POWER8 too uses KVM?
- Can we integrate POWER8 to OpenStack?
PCI Passthrough with KVM

Verification Environment:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>IBM Power System S822LC(Minsky)</td>
</tr>
<tr>
<td>OS</td>
<td>Ubuntu 17.10</td>
</tr>
<tr>
<td>KVM Version</td>
<td>1:2.10+dfsg-0ubuntu3.1</td>
</tr>
<tr>
<td>GPU</td>
<td>NVIDIA P100</td>
</tr>
<tr>
<td>Firmware</td>
<td>Upgrade to Beta Version</td>
</tr>
</tbody>
</table>

Fig. Passthrough with KVM
PCI Passthrough with KVM (Sample Output)

This gives you a Non-GPU virtual machine.

It is a script to create virtual machine
Virsh XML specification to run on Host

Create a domain from XML file
Connect to guest console
Connect to guest console
PCI Passthrough with KVM (Sample Output)

- Reusable image is created and CUDA has installed and tested.
- *.qcow2 which is under LABSVC/CUDA_VM/CUDA_VM-G.qcow2 has copied into image directory so that it can be reused.
- Now a virtual image can be repeatedly created from the “snapshot”

```
openstack@openstack:~/$LABSVC/CUDA_VM$ sudo virsh list
setlocale: No such file or directory

Id   Name       State
---  -------    ----
  3  NON_GPU_VM  running
 12  CUDA_VM    running
```

This gives you GPU virtual machine.

```
openstack@openstack:~/$LABSVC/tools$ /create_vm.sh -s G -n 4_GPU_VM -i 152 -c 4 -m 128 -d 30
creating 4_GPU_VM with ip 152
gpuspec is D gpuspec.spec
0: 0002:01:00:00
1: 0003:01:00:00
2: 000e:01:00:00
3: 000f:01:00:00
select all gpus you want to add (separated by space): 0 1 2 3
```

Users don’t need to install appropriate Nvidia-driver every time while creating VM.

More space will be needed.
PCI Passthrough with KVM (Sample Output)

- 4 GPUs is assigned to 4_GPU_VM and it is currently running.

```
openstack@openstack:~/$LABSVC/CUDA_VM$ sudo virsh list
setlocale: No such file or directory
Id   Name         State
---  -------      ----- 
   3  NON_GPU_VM   running
   11 4_GPU_VM     running
   12  CUDA_VM     running
```

4_GPU_VM is in running state

- And after login to 4_GPU_VM, we can see 4 NVIDIA GPU cards

```
ubuntu@4_GPU_VM:~$ lspci -nn | grep -i nvidia
00:08.0 3D controller [0302]: NVIDIA Corporation Device [10de:15f9] (rev a1)
00:0b.0 3D controller [0302]: NVIDIA Corporation Device [10de:15f9] (rev a1)
00:0e.0 3D controller [0302]: NVIDIA Corporation Device [10de:15f9] (rev a1)
00:11.0 3D controller [0302]: NVIDIA Corporation Device [10de:15f9] (rev a1)
```
Future work

OpenStack uses KVM

We verified that POWER8 can also use KVM

We can integrate POWER8 to OpenStack
Agenda

1. Our Background
2. Providing GPU Resources: P100
3. Benchmarking with different tool: DGX-1 vs S822LC for HPC(Minsky)
4. I/O advantages of IBM Power System
5. Performance improvement with Memory Interleave: nbody
6. Summary
Benchmarking with MatrixMulCuBLAS

Overload: MatrixA(12800,9600), MatrixB(9600,6400), MatrixC(12800,6400)

Higher is Better!!

- Since this program compares the performance of only one GPU, the performance is equivalent in both models equipped with the same GPU (Tesla P100)
- It contains simple matrix calculations so memory amount and bus width do not affect the results.

For both POWER8 and DGX-1, we found the same performance with 1 GPU

Overload: MatrixA(12800,9600), MatrixB(9600,6400), MatrixC(12800,6400)
Benchmarking with GROMACS

-It’s simulation software for benchmarking and measuring performance using different workloads such as proteins, lipids and nucleic acids.
**Benchmarking with Gromacs v5.1**

Workload: Comparison by sample simulation "3M Water Size" (water molecules)

- When using this program, DGX - 1 generally got high speed results as compare to POWER8 system for 1 GPU as well as for 4 GPUs.

- It is considered to be caused by the code of GROMACS v5.1 which is not optimized for Power system and also not optimized for softwares such as compiler, library, etc.

Graph. GPU performance comparison with Gromacs v5.1

Higher is Better!!
Benchmarkeding with Gromacs v2016.3

Workload: Comparison by sample simulation "6.8M Rnase_dodec data"

- POWER8 has more performance in case of 1 GPU, 2 GPUs and 4 GPUs than DGX-1 for GROMACS v2016.3

- GROMACS v2016.3 is optimized for POWER8 system.

- Fully SIMD CPU code for Power is available from this version

Graph. GPU performance comparison with Gromacs v2016.3
Benchmarking with nbody

- Nbody is kind of cuda sample program.
- This program can calculate single precision and double precision by using GPU and the results are displayed in GFLOPS.
- It can be also calculated by CPU only.

How to run nbody

```bash
$ ./nbody -benchmark -numbodies=2048000 -numdevices=1
```

- `-benchmark` : (run benchmark to measure performance)
- `-numbodies` : (number of bodies (>= 1) to run in simulation)
  (for GPU benchmark: 2048000, for CPU benchmark: 20480)
- `-numdevice` : (where i=(number of CUDA devices > 0) to use for simulation)
- `-cpu` : (run n-body simulation on the CPU)]
- `-fp64` : (use double precision floating point values for simulation)
Benchmarking with nbody

Workload: Comparison by numbodies=2048000, FP32

Higher is Better!!

- Although, performance of Power 8 increases up to 2 GPUs, but performance drops at single precision with 4 GPUs

Graph. GPU performance comparison with nbody for single and double precision.
Why such performance degradation and how to tackle it??
IBM lab Services has solved this problem, Please listen Part 2..
Thank you so much for listening.

Contact me at : a.purohit@ntt.com