Improving the Automation of Scaling-up an OpenHPC Cluster

(OpenHPC-Get-Mac Application)

Student:	Elham Hojati
OpenHPC Mentors:	Nirmala Sundararajan, Reese Baird
TTU Advisors:	Dr. Alan Sill, Dr. Yong Chen

https://github.com/nsfcac/Automating-the-scale-up-process-in-OpenHPC

IPMI

DMTF Redfish

GLEAMM

High Performance Computing Center

OUTLINE

- Background
  - IPMI
  - DMTF Redfish
  - OpenHPC
- The project: Improving the Automation of Scaling-up an OpenHPC Cluster
- Test Environment
- Implementation (OpenHPC-Get-Mac Application)
- Demo

Background

Baseboard Management Controller

- A baseboard management controller (BMC) is a dedicated processor inside the machine responsible for managing and monitoring the hardware layer of compute nodes, servers, or network devices.

- BMC performs these tasks through an individual independent connection.
  - IPMI
  - DMTF Redfish

IPMI

- The Intelligent Platform Management Interface (IPMI)
- It helps for monitoring and controlling the health and functionality of hardware.
  - independently of the host system's CPU, firmware (BIOS) and operating system.
- It is supported by more than 200 computer system vendors.
- IPMI runs on Baseboard Management Controller
- Typical features of an IPMI BMC
  - Hardware monitoring
  - Remote Control
    - Power Control

https://www.intel.com/content/www/us/en/servers/ipmi/ipmi-intelligent-platform-mgt-interface-spec-2nd-gen-v2-0-spec-update.html

DMTF’s Redfish™

- Redfish protocol was designed as an open industry standard to meet scalability requirements.

- It uses **RESTful interfaces** to perform operations and **JSON and OData formats** for data payloads.

- DMTF’s Redfish™ is an open industry standard specification and schema.
- Redfish is defined as a standard and a RESTful API for the management of scale-out commodity servers.
- its goal : addressing all the components in the data center with a consistent API.

https://www.dmtf.org/standards/redfish

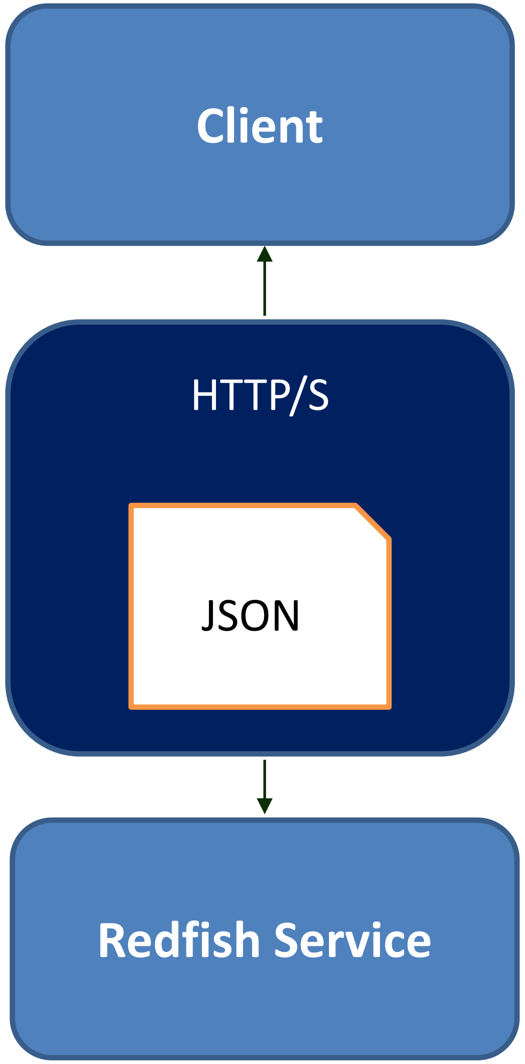
DMTF Redfish Technology

Redfish Interface (RESTful)

- Request/response via HTTP/HTTPS protocol
- Content is formatted in JSON
- Schemas available to describe JSON

Redfish Models

- Models for managing compute, storage and network platforms and services



https://www.dmtf.org/standards/redfish

OpenHPC

- OpenHPC is a HPC-centric software for creating, composing, and administering high-performance computing clusters.

- It is a collection of open-source HPC software components, methods and tools.

- OpenHPC provides methods to register nodes and provide the information needed to provision them and integrate them into data centers.

- In the current OpenHPC methodology for creating a cluster, we need to have the Mac addresses of all the compute nodes in advance and manually type in the "input.local" file. It is a time-consuming step, especially for big data centers with a massive number of compute nodes.

The project:
Improving the Automation of Scaling-up an OpenHPC Cluster

Improving the Automation of Scaling-up an OpenHPC Cluster

- This project aims to use IPMI and Redfish to get access to the BMC of the nodes and gather the MAC addresses based on the selected internal interface by the admin of the cluster.

- It helps to automate the process of adding nodes to a data center better than it has been.

- It uses the BMC to explore the Mac addresses of the interfaces and allow the user to select which will be the provisioning interface, which will be the control interfaces.

The Tool Features

Network discovery:

- It gets a range of BMC-IP addresses and checks the list of available network interfaces. It saves the information in a JSON file.

Expanding the cluster:

- It gets a range of BMC-IP addresses, and based on the internal network NIC, it updates the “input.local” file by inserting the mac-addresses of nodes there.

Test Environment: Zephyr Cluster, RedRaider Cluster

TEST ENVIRONMENT

- The Global Laboratory for Energy Asset Management and Manufacturing (GLEAMM)
  - Combines the research and commercialization expertise of Texas Tech University with the field testing, certification and development expertise of Group NIRE, a for-profit energy development company.

https://gleamm.org

ZEPHYR CLUSTER

In Rack 91:

31 compute node: powerEdge R410

1 storage node with 12 of 1 tb SAS (12 TB)

In Rack 92:

30 compute node: powerEdge R410

1 storage node with 12of 6 tb SAS (72 TB)

4 compute node: powerEdge C6320

In Rack 93:

33 compute node: powerEdge R410

powerEdge C6320 info:

System memory size: 128 GB

2 (Intel(R) Xeon(R) CPU E5-2698 v3 @ 2.20 GHz) processors , 16 cores per processor

powerEdge R410 info:

System memory size: 24 GB (EDC DDR3)

2 (Intel(R) Xeon(R) CPU X5660 @ 2.80 GHz) processors , 6 cores per processor

Overall we have 1256 cores, and 2.768 TB of memory, and 84 TB of storage.

HPE G2 rack PDUs : Redfish Enabled

Zephyr Cluster Rack Map

	Zephyr Cluster		
Rack # /	91	92	93
Slot in Rack			
42			RPS-600
41	1GB Eth Switch	1GB Eth Switch	1GB Eth Switch
40		Core SW	
39	R410	Core SW	R410
38	R410	R410	R410
37	R410	R410	R410
36	R410	R410	R410
35	R410	R410	R410
34	R410	R410	R410
33	R410	R410	R410
32	R410	R410	R410
31	R410	R410	R410
30	R410	R410	R410
29	R410	C6320	R410
28	R410	(4 Nodes)	R410
27		NFS Server	
26		Ancell OSS2 (72 TB)	
25	IB-12300	IB-12300	
24	IB-12300	IB-12300	
23	R410	R410	R410
22	R410	R410	R410
21	R410	R410	R410
20	R410	R410	R410
19	R410	R410	R410
18	R410	R410	R410
17	R410	R410	R410
16	R410	R410	R410
15	R410	R410	R410
14	R410	R410	R410
13	R410	R410	R410
12	R410	R410	R410
11		R410	
10	R410	R410	R410
9	R410	R410	R410
8	R410	R410	R410
7	R410	R410	R410
6	R410	R410	R410
5	R410	R410	R410
4	NFS Server	R410	R410
3	(12 TB)	R410	R410
2			
1			

GLEAMM FACILITIES

Zephyr Cluster: Consumer	Gleamm Microgrid: Energy Provider
A data center (3 Racks): ▪ 94 compute nodes ▪ 2 storage nodes 	▪ Solar Plant ▪ Outback Battery ▪ Outback Inverter ▪ Diesel Generator ▪ Grid ▪ Load Banks 

The Second Testbed

The **RedRaider** Cluster in High-Performance Computing Center at TTU.

- It has Redfish-Enabled compute nodes.

https://www.depts.ttu.edu/hpcc/

Implementation: OpenHPC-Get-Mac Application

About the OpenHPC-Get-Mac Application

The OpenHPC-Get-Mac Application: https://github.com/nsfcac/Automating-the-scale-up-process-in-OpenHPC

Code:

- ipmi.py
- displayInfoIPMI.py
- redfish.py
- displayInfoRedfish.py
- userInterfaceApp.py

Input:

- Readme
- credentialInfo.txt
- clusterInfo

Output:

- ClusterNetInfo.json
- input.local

How To use the application

Step 1: Make sure you have the following python libraries.

- requests
- ipaddress
- json
- datetime
- subprocess
- codecs
- Thread
- time
- os

How To use the application

Step 2: Set up input files.

- Modify clusterInfo file in the Doc-Files directory, and insert the BMC IP addresses of the nodes there(one IP per line).
- Modify the credentialInfo.txt file in the Doc-Files directory, and set up the BMC-User and BMC_Password of the cluster in this file.

Step 3: Run the OpenHPC-Get-Mac application.

Run the OpenHPC-Get-Mac application using this command.

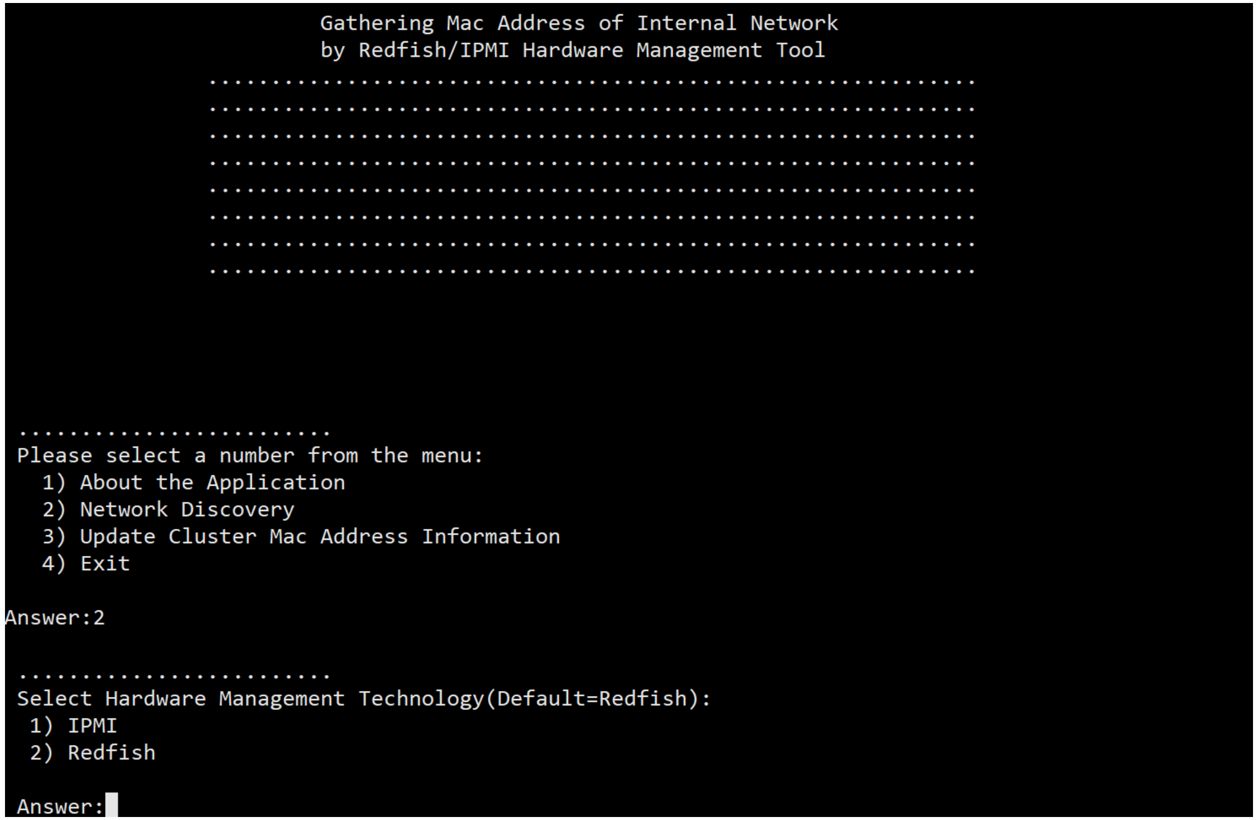
- python3 userInterfaceApp.py

How To use the application

Step 4: Follow the process based on the interface guideline.

The application gives you four options:

1) About the Application
2) Network Discovery
3) Update Cluster Mac Address Information
4) Exit



Network Discovery

```
---------------------------------------------

BMC_IP: 10.101.6.1
BMC Hostname: cpu-6-1
Hostname: None
Timestamp: 2021-01-27 04:20
Power On: True
NIC Mac Addresses information:
NIC_Id :  NIC.Embedded.1-1-1
MACAddress :  7C:D3:0A:C6:39:60
NIC_Id :  NIC.Embedded.2-1-1
MACAddress :  7C:D3:0A:C6:39:62

---------------------------------------------

BMC_IP: 10.101.3.10
BMC Hostname: cpu-3-10
Hostname: None
Timestamp: 2021-01-27 04:20
Power On: True
NIC Mac Addresses information:
NIC_Id :  NIC.Embedded.1-1-1
MACAddress :  7C:D3:0A:D7:D1:08
NIC_Id :  NIC.Embedded.2-1-1
MACAddress :  7C:D3:0A:D7:D1:0A

---------------------------------------------

BMC_IP: 10.101.4.3
BMC Hostname: cpu-4-3
```

```
---------------------------------------------

BMC_IP: 10.101.92.8
Hostname: None
Timestamp: 2021-01-27 02:59
NIC Mac Addresses information:
MACAddress :  78:2b:cb:2f:be:20
MACAddress :  78:2b:cb:2f:be:21

---------------------------------------------

BMC_IP: 10.101.92.6
Hostname: None
Timestamp: 2021-01-27 02:59
NIC Mac Addresses information:
MACAddress :  84:2b:2b:48:31:8b
MACAddress :  84:2b:2b:48:31:8c

---------------------------------------------

BMC_IP: 10.101.92.2
Hostname: None
Timestamp: 2021-01-27 02:59
NIC Mac Addresses information:
MACAddress :  84:2b:2b:55:12:6b
MACAddress :  84:2b:2b:55:12:6c
```

Update Cluster Mac Address Information

```
Please select a number from the menu:
  1) About the Application
  2) Network Discovery
  3) Update Cluster Mac Address Information
  4) Exit

Answer:3

 .........................
Select Hardware Management Technology(Default=Redfish):
 1) IPMI
 2) Redfish

Answer:1

 .........................
Select Network Interface?(Default=NIC1):
  1)NIC1
  2)NIC2
  3)NIC3
  4)NIC4

Answer:1

 .........................
Select compute_prefix(Default=c):

Answer:zc-92-

 .........................
Select the path to the cluster information file (Default: ../Doc-Files/clusterInfo):
```

Output

- ../Doc-Files/ClusterNetInfo.json file.
- ../Doc-Files/ input.local file

```
BMC_IP: 10.101.92.8
Hostname: c18
Timestamp: 2021-01-27 05:27
NIC Mac Addresses information:
MACAddress :  78:2b:cb:2f:be:20
MACAddress :  78:2b:cb:2f:be:21

--------------------------------------------

BMC_IP: 10.101.92.10
Hostname: c20
Timestamp: 2021-01-27 05:27
NIC Mac Addresses information:
MACAddress :  84:2b:2b:48:28:c2
MACAddress :  84:2b:2b:48:28:c3

--------------------------------------------

BMC_IP: 10.101.92.7
Hostname: c17
Timestamp: 2021-01-27 05:27
NIC Mac Addresses information:
MACAddress :  84:2b:2b:48:31:eb
MACAddress :  84:2b:2b:48:31:ec
```

```
NIC_Id :  NIC.Embedded.2-1-1
MACAddress :  7C:D3:0A:D7:C3:54

----------------------------------------------

BMC_IP: 10.101.3.6
BMC Hostname: cpu-3-6
Hostname: compute-26
Timestamp: 2021-01-27 04:23
Power On: True
NIC Mac Addresses information:
NIC_Id :  NIC.Embedded.1-1-1
MACAddress :  7C:D3:0A:D0:D8:22
NIC_Id :  NIC.Embedded.2-1-1
MACAddress :  7C:D3:0A:D0:D8:24

----------------------------------------------

BMC_IP: 10.101.6.6
BMC Hostname: compute-6-6
Hostname: compute-56
Timestamp: 2021-01-27 04:23
Power On: True
NIC Mac Addresses information:
NIC_Id :  NIC.Embedded.1-1-1
MACAddress :  7C:D3:0A:C6:3F:12
NIC_Id :  NIC.Embedded.2-1-1
MACAddress :  7C:D3:0A:C6:3F:14
```

ClusterNetInfo.json

ClusterNetInfo.json


```
    },
    "systemName": "System.Embedded.1",
    "HN-BMC": "cpu-2-5",
    "power state": true
}


{
    "BMC_IP": "10.101.3.2",
    "hostname": "compute-22",
    "time": "2021-01-27 04:23",
    "NICs": {
        "NIC1": {
            "NIC_Id": "NIC.Embedded.1-1-1",
            "URI": "/redfish/v1/Systems/System.Embedded.1/EthernetInterfaces/NIC.Embedded.1-1-1",
            "MACAddress": "7C:D3:0A:D0:E2:90"
        },
        "NIC2": {
            "NIC_Id": "NIC.Embedded.2-1-1",
            "URI": "/redfish/v1/Systems/System.Embedded.1/EthernetInterfaces/NIC.Embedded.2-1-1",
            "MACAddress": "7C:D3:0A:D0:E2:92"
        }
    },
    "systemName": "System.Embedded.1",
    "HN-BMC": "cpu-3-2",
    "power state": true
}
```

input.local

```
compute_prefix="${compute_prefix:-com
sms_eth_internal="${sms_eth_internal:
c_name[0]=compute-1
c_name[1]=compute-15
c_name[2]=compute-22
c_name[3]=compute-14
c_name[4]=compute-45
c_name[5]=compute-39
c_name[6]=compute-31
c_name[7]=compute-37
c_name[8]=compute-5
c_name[9]=compute-3
c_name[10]=compute-19
c_name[11]=compute-36
c_name[12]=compute-57
c_name[13]=compute-43
c_name[14]=compute-9
c_name[15]=compute-24
c_name[16]=compute-16
c_name[17]=compute-4
c_name[18]=compute-35
c_name[19]=compute-11
c_name[20]=compute-50
c_name[21]=compute-28
c_name[22]=compute-13
c_name[23]=compute-38
c_name[24]=compute-6
c_name[25]=compute-17
c_name[26]=compute-21
c_name[27]=compute-44
c_name[28]=compute-8
c_name[29]=compute-60
```

Input.local contains:

- compute_prefix
- sms_eth_internal
- c_name[i]
- c_bmc[i]
- c_mac[i]

Demo

Demo

Demo-paer1: Running the Application in the RedRaider Cluster (using Redfish)

- **Testbed: RedRaider Cluster**

Demo-part2: Running the Application in the Zephyr Cluster (using IPMI)

- **Testbed: Zephyr Cluster**

Demo Link:
https://drive.google.com/file/d/1rSuk2ugNkjkiPHGa4pc7ef5pP9V6tV05/view?usp=sharing

Presentation Link:
https://drive.google.com/file/d/1f3Yd5aaDj9zhgw_8wExsCsIUUdvJE7s5/view?usp=sharing

File Edit View History Bookmarks Tools Help

nsfcac/Automating-the-scale-u

https://github.com/nsfcac/Automating-the-scale-up-process-in-OpenHPC 110%

Search or jump to... / Pull requests Issues Marketplace Explore

Learn Git and GitHub without any code!

Using the Hello World guide, you'll start a branch, write comments, and open a pull request.

Read the guide

nsfcac / **Automating-the-scale-up-process-in-OpenHPC**

Unwatch 5 | Star 0 | Fork 1

Code | Issues | Pull requests | Discussions | Actions | Projects | Wiki | Security | Insights | Settings

master | 1 branch | 0 tags | Go to file | Add file | Code

elham1296 6th Commit	030c4d7 6 days ago	14 commits
Codes	My Third Commit	6 days ago

About

Improving the Automation of Scaling-up an OpenHPC Cluster (Gathering Mac Address of Internal Network by Redfish/IPMI Hardware Management Tools) OpenHPC is a tool for creating