



Centralized Node Attribute Database for High Performance Computing

Nisha Prabhakar
Meghan Utter

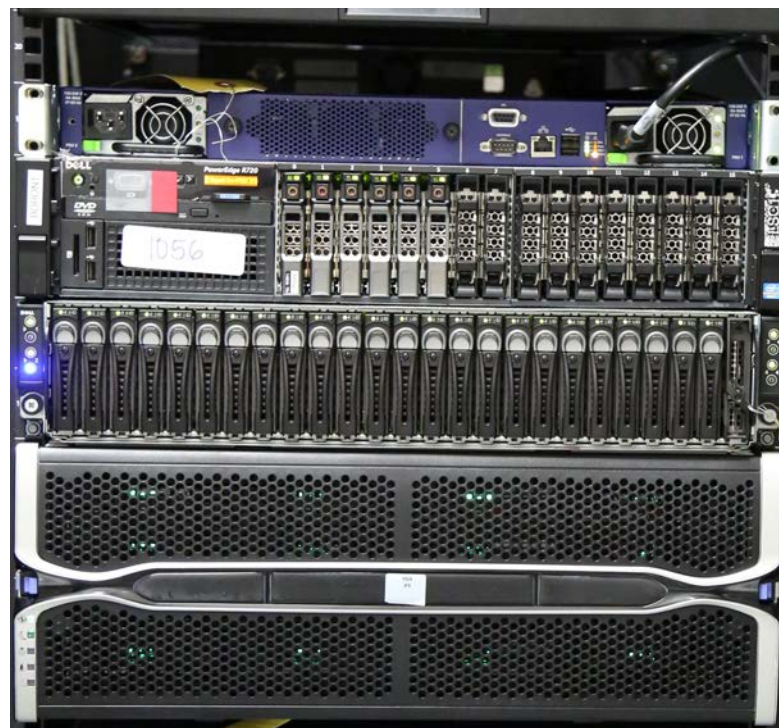
Computing/COMP-LC/HPC Cluster Engineer Academy





Motivation

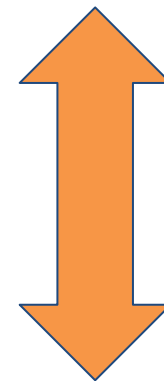
- The *genders* tool is an open source LLNL tool that stores information about node configurations
- There is no way to access this information without logging into a node
- -> we created a *centralized database* which stores all the genders of all the clusters





Approach

1. Installed the genders library for python in python3
2. created the structure of the database
3. used python3 to populate the database
 - a. adapted the script to comb through multiple directories
4. created python methods for users to query from the database





Results and Conclusions

root@boron2:~

```
[root@boron2 ~]# nodeattr -q login
[root@boron2 ~]# python3 genBase.py3 -q login
direct [2-3]
[root@boron2 ~]#
```

- nodeattr -q queries the local file, while our script queries the database in a similar manner
- *who does this benefit:* system engineers
- *what's next*
 - use gitlab to automate updates to the database
 - implement the database in larger clusters (integrate with cfengine structure)
 - ideally, any cluster would be able to use our query script to query the central node



Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.



Implementing and Interfacing with KVM

Cristian Palomo-Ramirez
cpalomo-ramirez@csumb.edu

Ben Ryan
bpryan@email.wm.edu

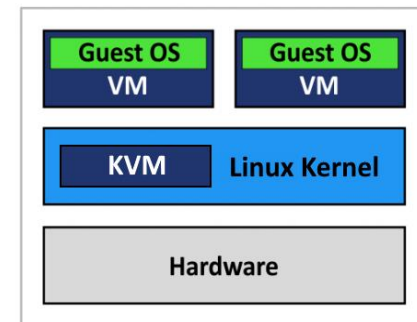
HPC Cluster Engineering Academy



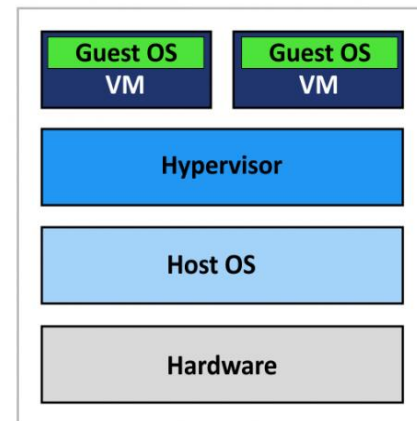
What is KVM and why is it useful?



- A Kernel-based Virtual Machine (KVM) is a type of VM that turns the Linux kernel into a bare-metal hypervisor.
- KVM allows the host machine to treat every guest (VM) as if it were a Linux process.
- Some of the benefits of a bare-metal VM are efficient usage of resources for smaller, specialized tasks, ease of testing, etc.
- The main benefits of KVM specifically is that it is built into Linux and is extremely efficient.



Type 1 Hypervisor
(Bare-Metal Architecture)



Type 2 Hypervisor
(Hosted Architecture)

Modified from Nakivo.com



Implementation and Tools

- Virt-manager is a GUI tool commonly used to manage KVM guests
- libvirt is the tool/package for managing guests via the command line
- Installing a variety of Linux distributions manually on individual guests is time consuming.
- Using tools such as BASH scripting and Ansible can automate this process.
 - Kickstart allows for pre-configuration of the operating system prior to installation.



High Level Goals

- Set up environment for testing MSR-safe kernel modules
- Create base images for various Linux distributions with configuration
 - Required packages
 - Test user with sudo privileges
 - Automated installation
 - Documentation for future admin use/maintenance
- Respond to Gitlab Continuous Integration requests
 - script/tool to run the CI request on the allocated image
 - capture results/logs
 - deallocate/clean up instance



Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.