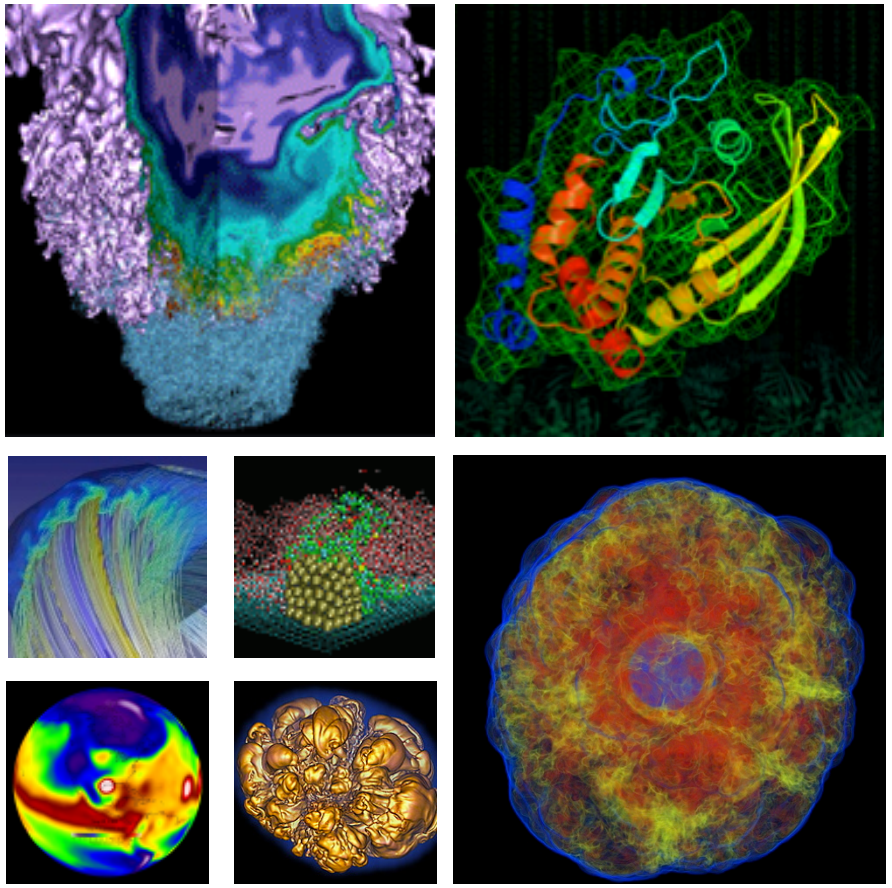
Shifter Overview



Shane Canon

CUG17 - Tutorial

Topics

- Shifter Architecture and Design
- Shifter in Action
- Discussion and Future Work

Why not just run Docker

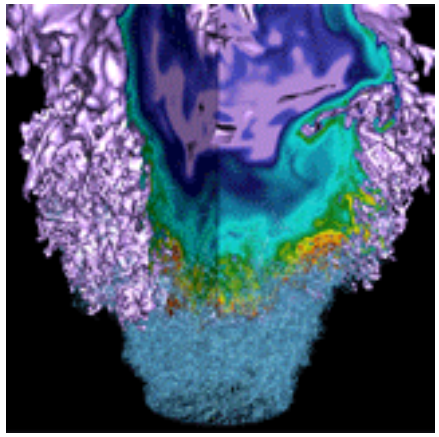
- System Architecture: Docker assumes local disk
- Security: Docker currently uses an all or nothing security model. Users would effectively have system privileges
- Integration: Docker doesn't play nice with batch systems.
- System Requirements: Docker typically requires very modern kernel
- Complexity: Running real Docker would add new layers of complexity

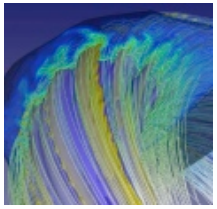
```
> docker run -it -v /:/mnt --rm busybox
```

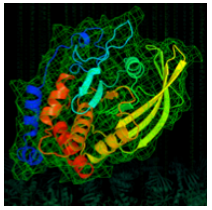
Solution: Shifter

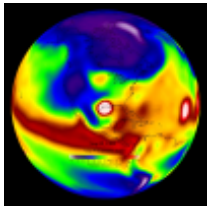
- **Design Goals:**
  - User independence: Require no administrator assistance to launch an application inside an image
  - Shared resource availability (e.g., file systems and network interfaces)
  - Leverages or integrates with public image repos (i.e. DockerHub)
  - Seamless user experience
  - Robust and secure implementation
- **Hosted at GitHub:**
  - https://github.com/nersc/shifter

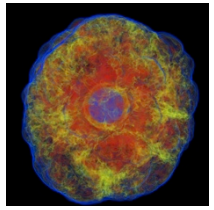
Implementation

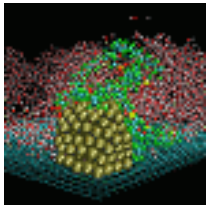






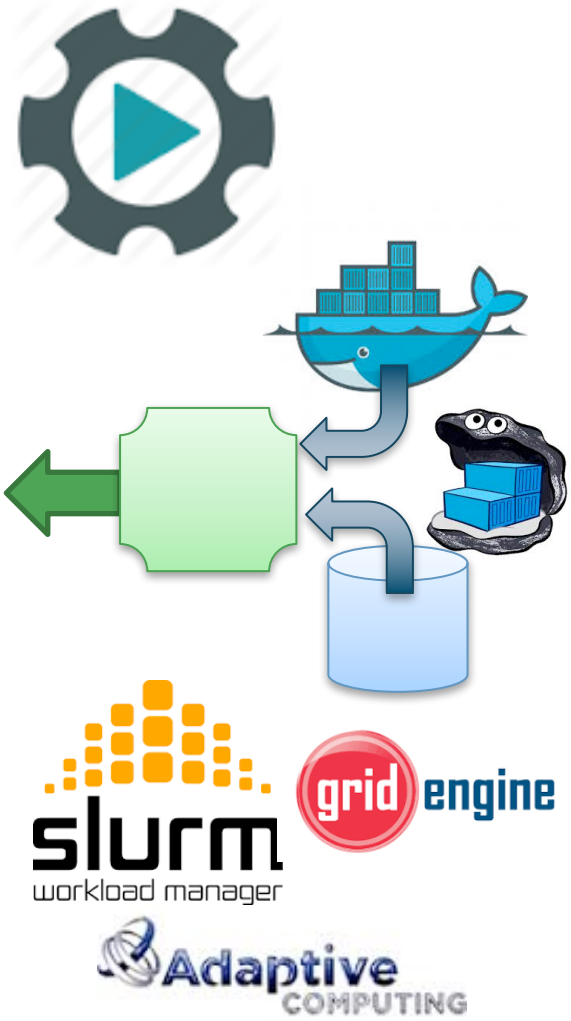




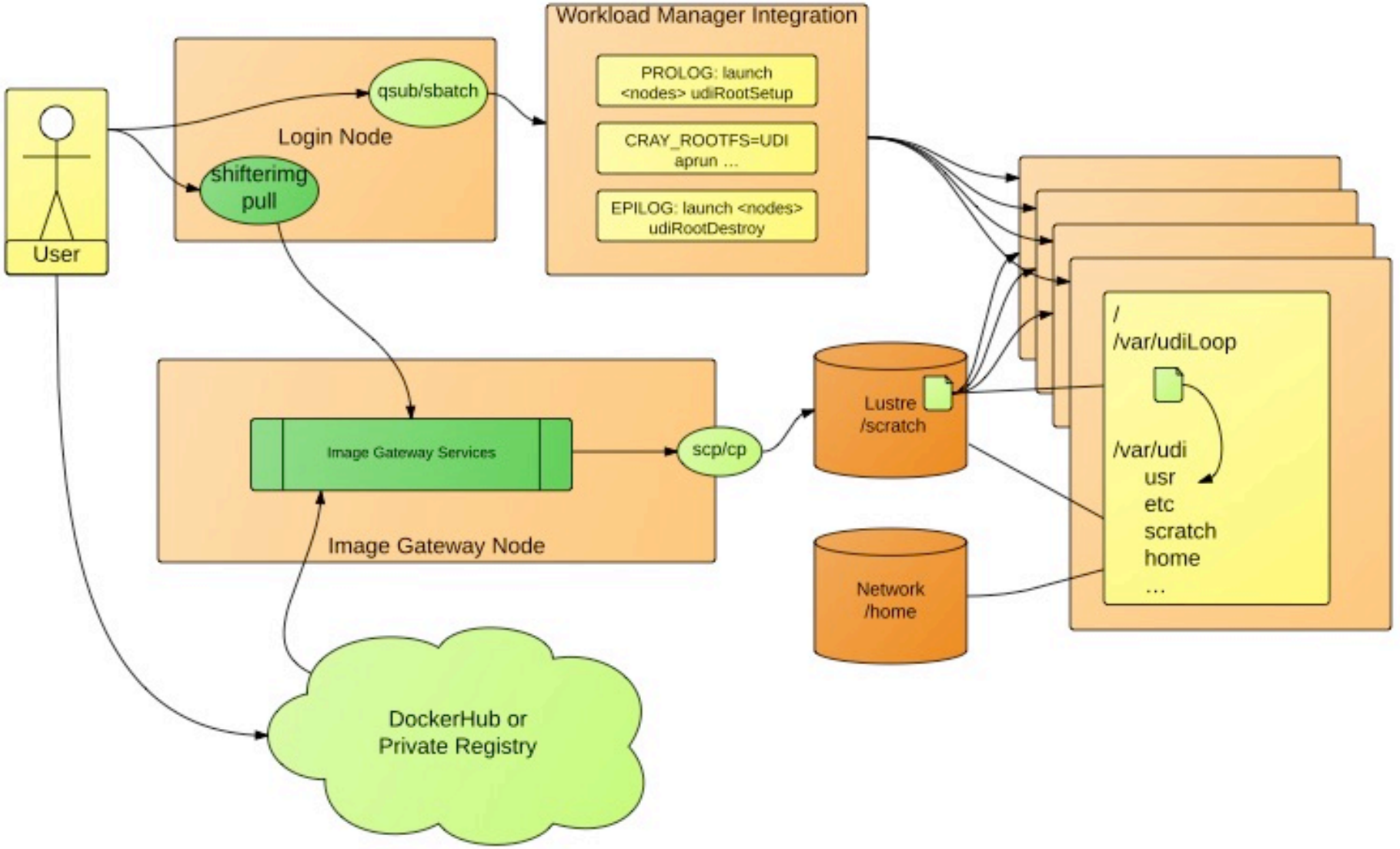


Shifter Components

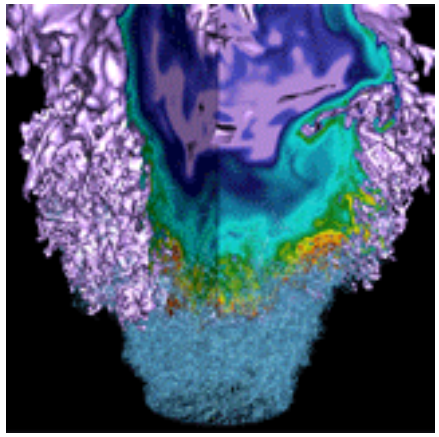
- **Shifter Image Gateway**
  - Imports and converts images from DockerHub and Private Registries
- **Shifter Runtime**
  - Instantiates images securely on compute resources
- **Work Load Manager Integration**
  - Integrates Shifter with WLM

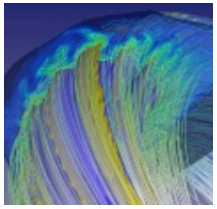


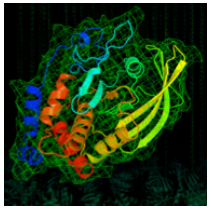
Shifter Architecture and Flow

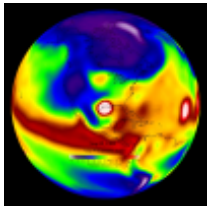


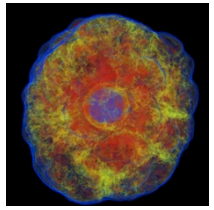
Shifter in Action

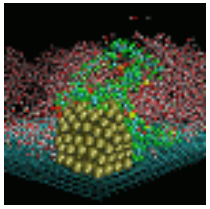




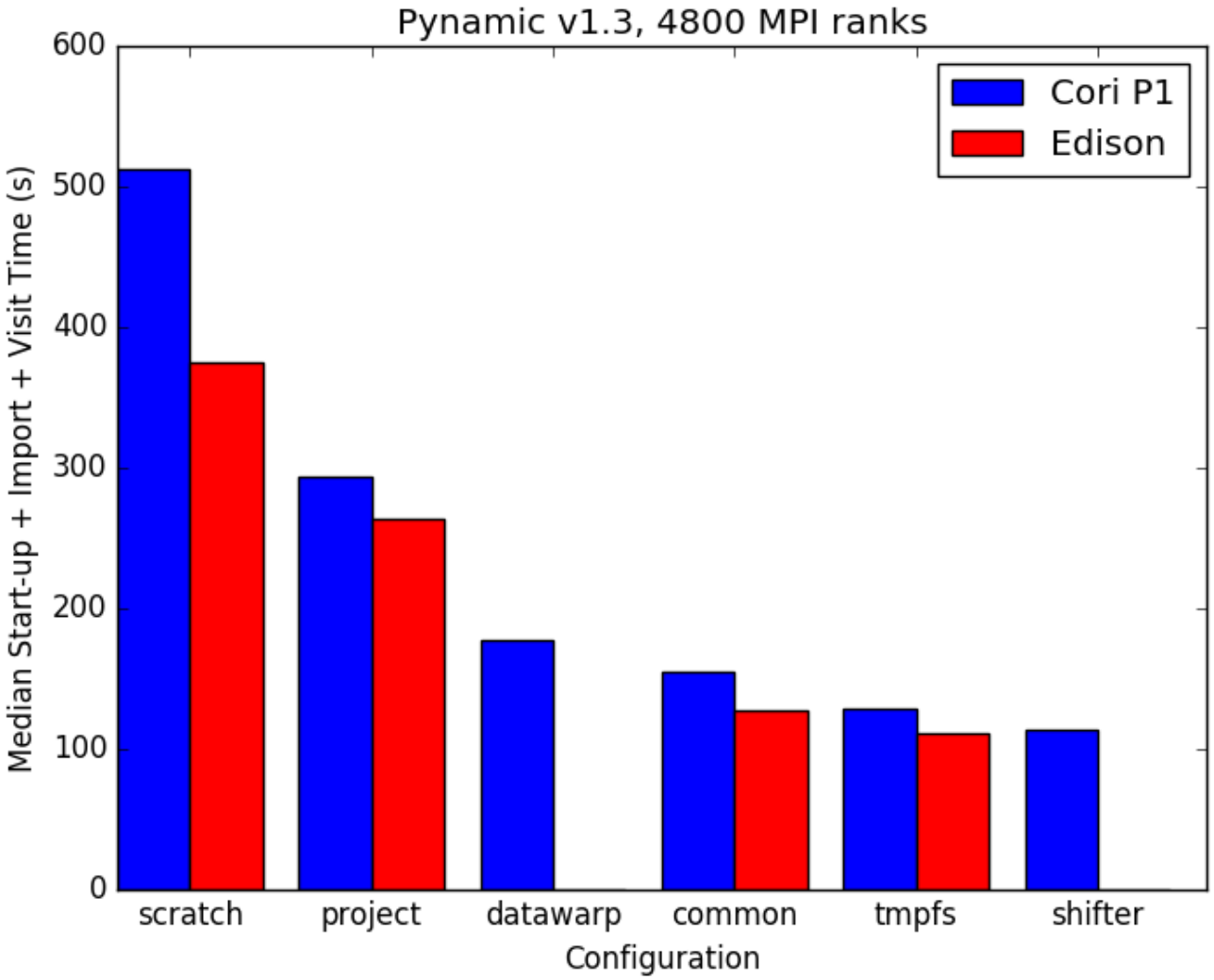








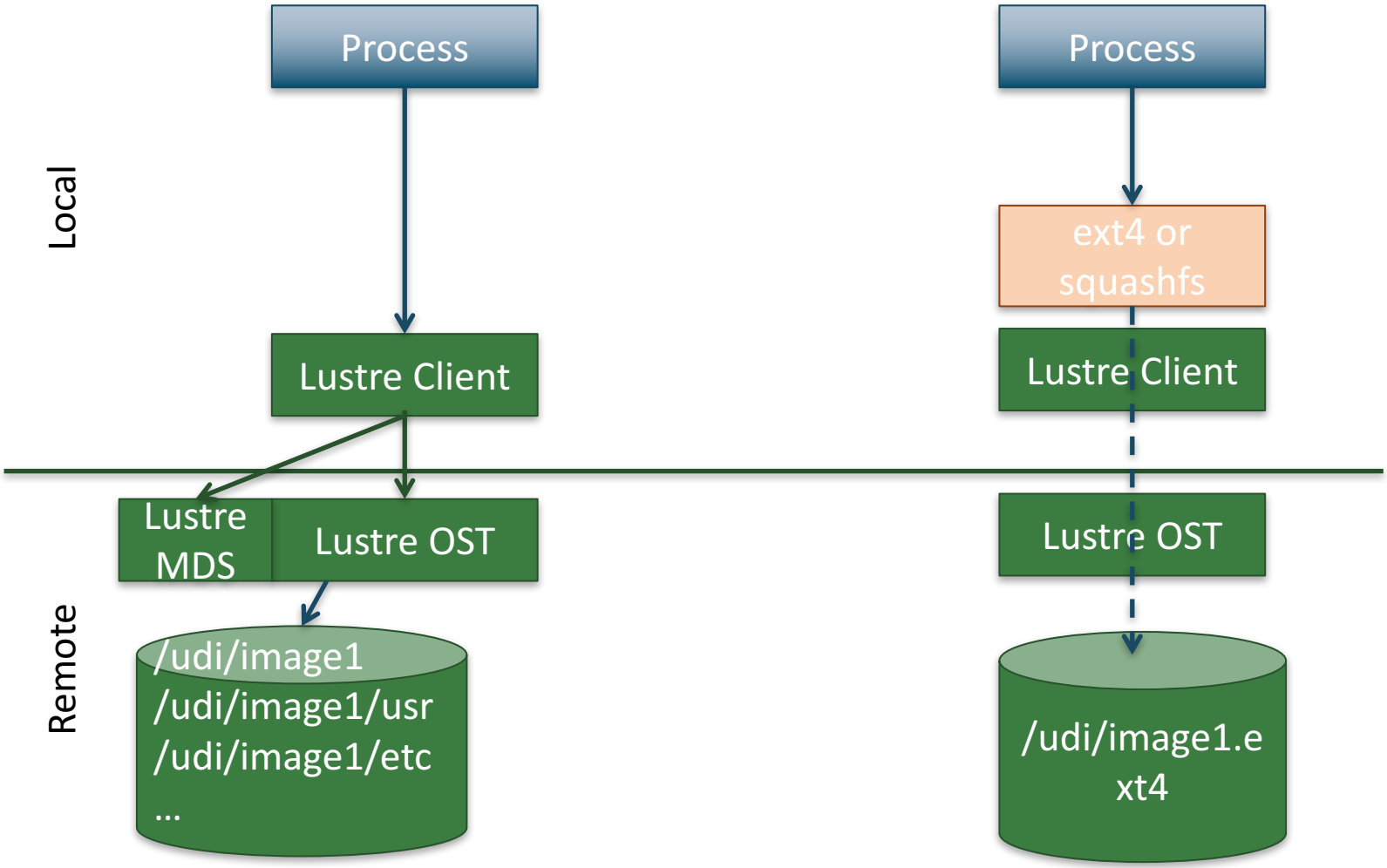
Shifter accelerates Python Apps



Why?

- Python must walk through the python libraries to construct the namespace
- Python must load up (read) any dynamic libraries that are required
- The loader must traverse the LD_LIBRARY_PATH to find the libraries to load
- Result: Lots of metadata accesses which put a load on the file system Metadata server

File System flow – Traditional vs Shifter



Per-Node Write Cache

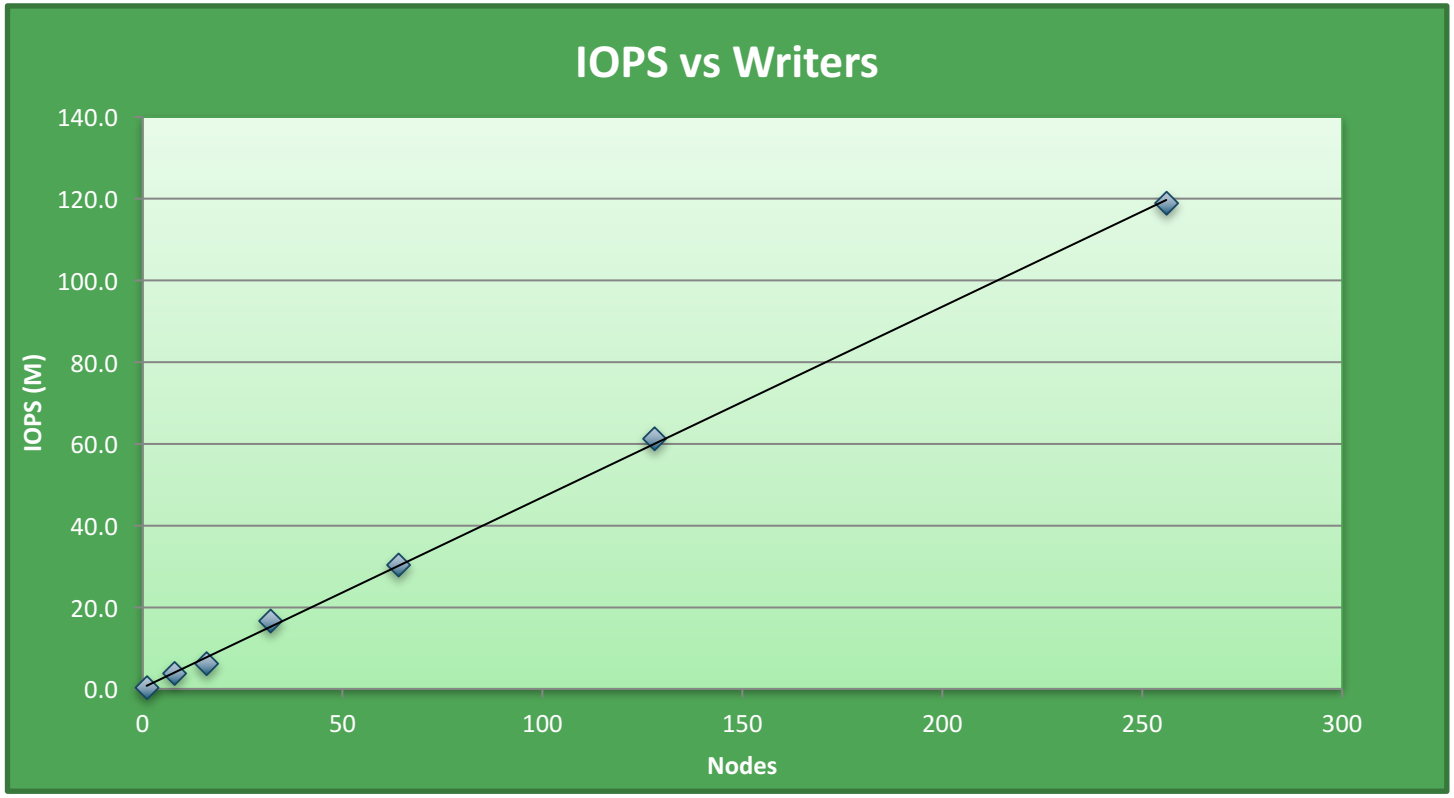
Per-Node Write Cache provides local disk like functionality but is backed by the Parallel File System.

Nodes/Writers per node	Lustre (MB/s) per writer	Shifter (MB/s) per writer	Real Local Disk (MB/s) per writer
1/1	83	594	416
10/10	87	625	416*
10/20	67	616	165*
10/40	55	589	53*
20/40	71	627	165*
20/80	55	588	53*

Results of a simple "dd" test to simulate writing ~5GB of small transaction I/O (**`dd if=/dev/zero of=$TARGET bs=512 count=10M`**)

* Extrapolated from a single node test

Per-Node Write Cache (IOPS)



Results of an IOR File per-process, 2 tasks per node, 512B transfer size, 2GB write. 100x faster than Lustre at the same scale.

Shifter and MPI

- **In Image**
  - Add required libraries directly into image.
  - Users would have to maintain libraries and rebuild images after an upgrade.
- **Managed Base Image (Golden Images)**
  - User builds off of a managed image that has required libraries.
  - Images are built or provided as part of a system upgrade.
  - Constrained OS choices and a rebuild is still required.
- **Volume Mounting**
  - Applications built using ABI compatibility.
  - Appropriate libraries are volume mounted at run time.
  - No rebuild required, but may not work for all cases.

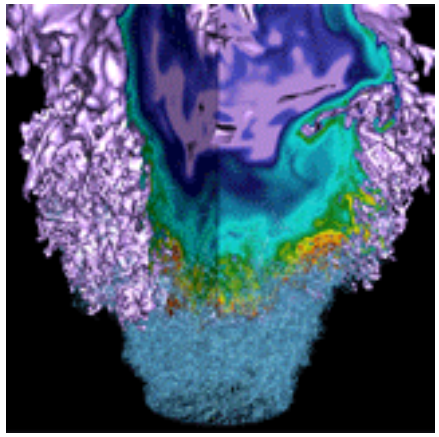
Volume Mount Approach

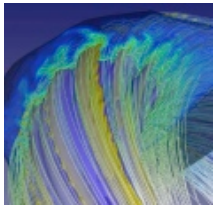
```
FROM nersc/mpi-ubuntu:14.04

ADD . /app
RUN cd /app && \
    mpicc –o hello helloworld.c
```

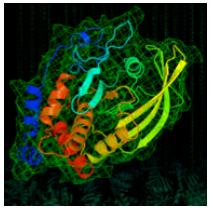
Dockerfile

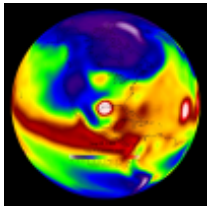
```
> docker build –t scanon/hello-vm:latest .
> docker push scanon/hello-vm:latest
```

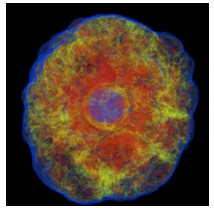
Discussion and Future Work

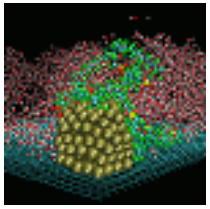












Why Users will like Shifter

Enables regular users to take advantage of Docker on HPC systems at scale.

This enables users to:

- **Develop an application on the desktop or laptop and easily run it on a cluster or Supercomputer**
- **Solve their dependency problems themselves**
- **Run the (Linux) OS of their choice and the software versions they need**

And…

- **Improves application performance in many cases**
- **Improves reproducibility**
- **Improves sharing (through sites like Dockerhub)**

How does Shifter differ from Docker?

Most Noticeable

- **Image read-only on the Computational Platform**
- **User runs as the user in the container – not root**
- **Image modified at container construction time (e.g. additional mounts)**

Less Noticeable:

- **Shifter only uses mount namespaces, not network or process namespaces**
- **Shifter does not use cgroups directly (integrated with the Workload Manager)**
- **Shifter uses individual compressed filesystem files to store images, not the Docker graph (slows down iterative updates)**
- **Shifter starts some additional services (e.g. sshd in container space)**

Roadmap

- **16.08 Release:**
  - Support for RHEL 6/7, SLES 11/12, Rhine/Redwood
  - RPM builds
  - Improved scaling
  - UI Improvements
  - Per-node write cache
  - Security and stability improvements
  - Image expiry and removal
- **17.0? Release**
  - ACL support (private and authenticated images)
  - Image usage statistics and metrics
  - GPU Support (CSCS)
  - Bug Fixes

Conclusions

- **Shifter enables HPC systems to run Containers easily and at large scale**
- **Shifter provides the flexibility of Docker without sacrificing security, scalability or performance.**
- **Shifter opens the door to the many benefits of Docker including easy sharing of images, reproducibility, etc.**

Questions?