

MVPNet

TESSI

dThreads

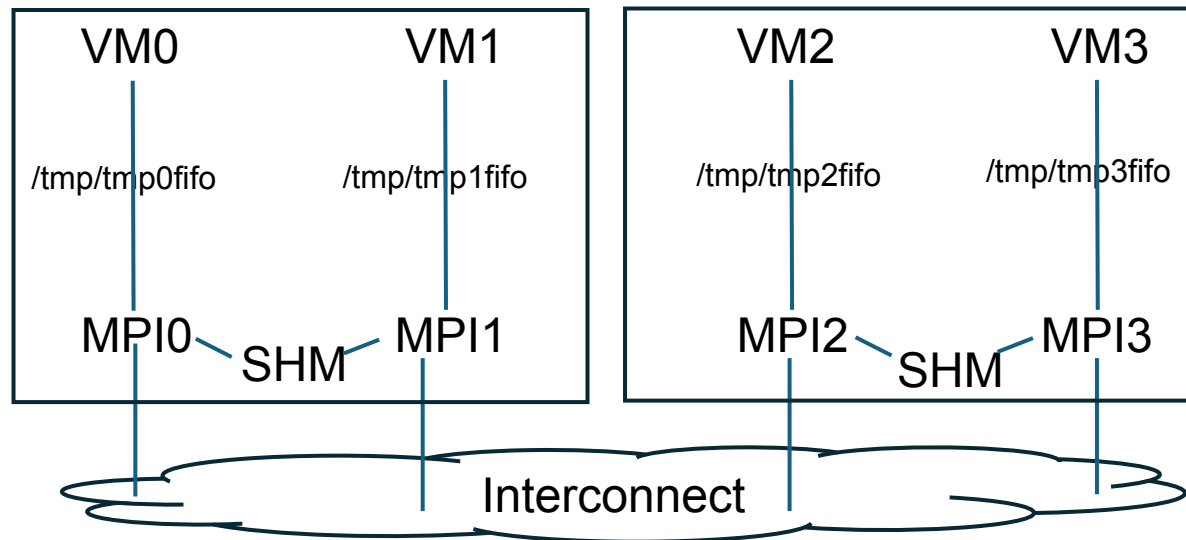
MSST 2026
John Bent

Integration challenges

- Testbeds are expensive
- At scale testbeds are prohibitively expensive
 - Only current known solution: re-use old systems
- Another approach?
 - Can we emulate alternative systems on production machines running as purely a regular user

Emulate Alternative System on Production System?

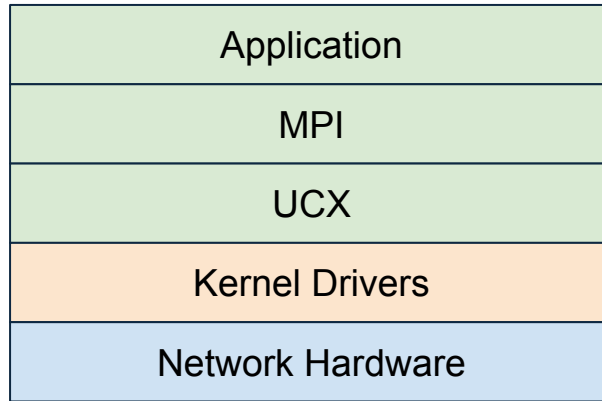
- One challenge is network
- Could we use MPI to create VMs?
 - Connect VMs via MPI messaging



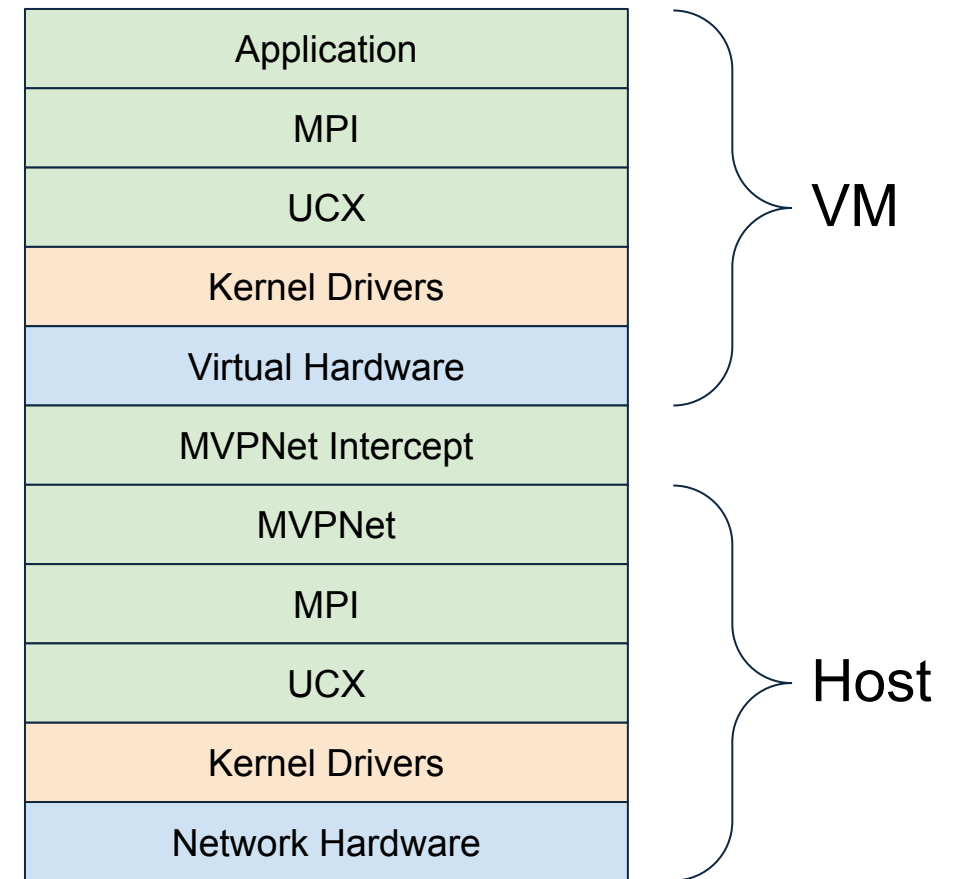
Introducing MVPNet!

EMC3 Project
currently being
co-developed by
CMU and LANL

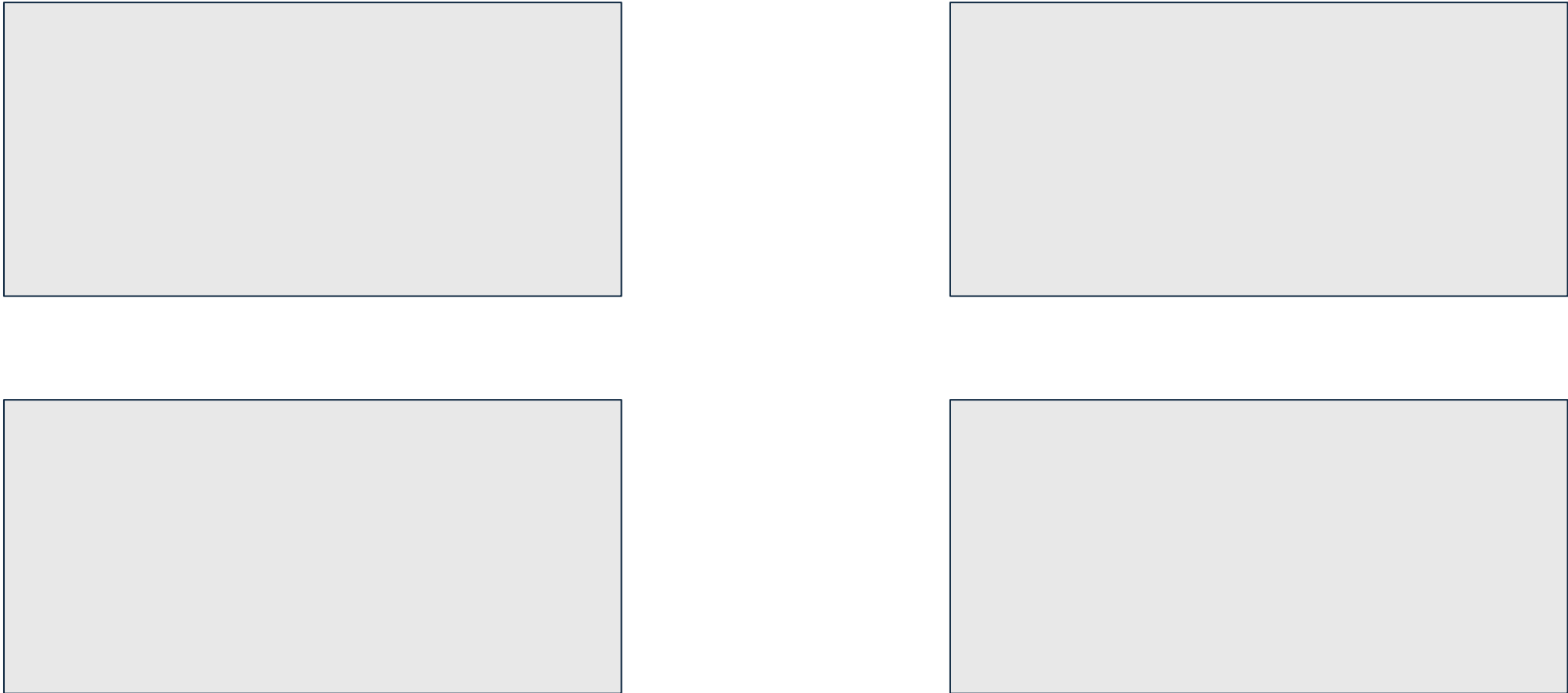
Normal MPI Stack



MVPNet Stack

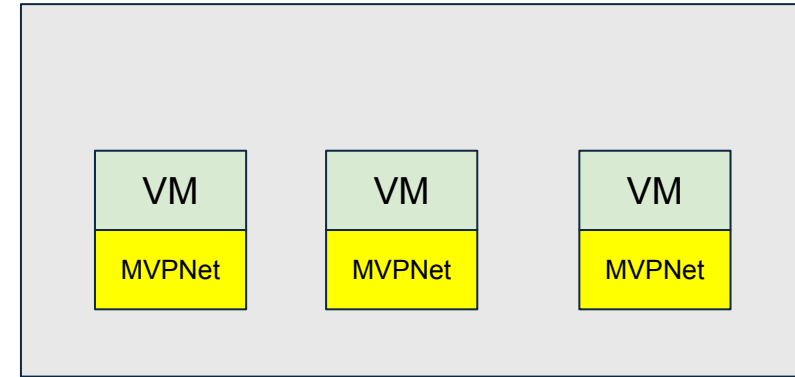
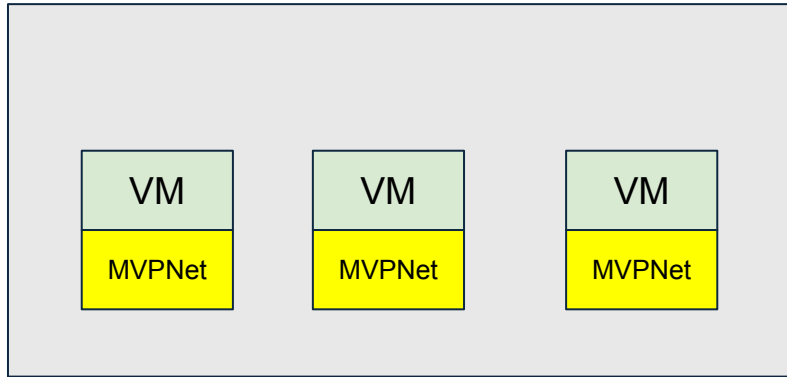


Virtual Clusters with MVPNet

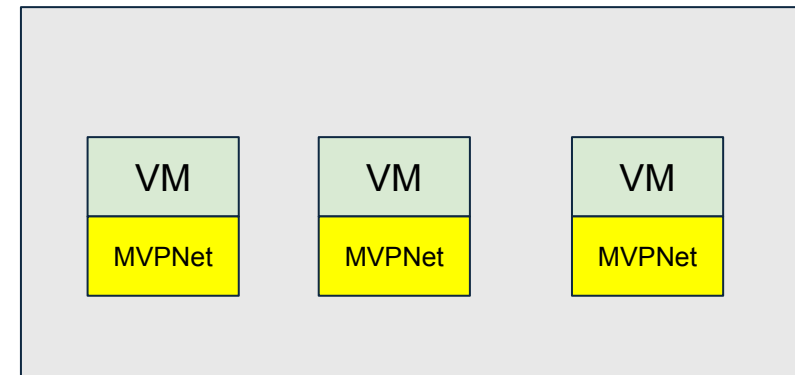
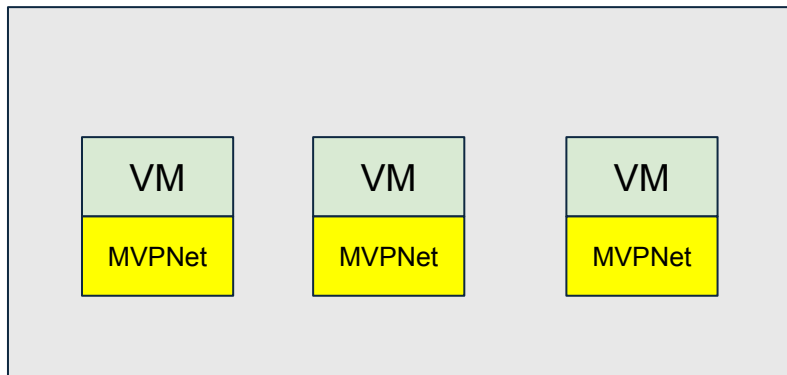


Four physical nodes where user does not have root privilege

Virtual Clusters with MVPNet



12 interconnected VMs where user does have root



Four physical nodes where user does not have root privilege

Key Advantages of MVPNet

1. User-space
 - a. Note that root is available within the VMs
2. Optimized messaging
 - a. MPI's *raison d'être* is optimized transparent networking
 - b. Shmem on-node, IB, Slingshot, etc all accessible via simple API

Exemplar Usage of MVPNet: TESSI

So now you have a bunch of virtual machines in a virtual network connected transparently via MPI.

Now what?

TESSI

Tool for Extensible Scalable Storage Infrastructure

How Can We Improve the Lives of Current Admins?

TESSI: Tool for Agile Scalable Storage Infrastructure

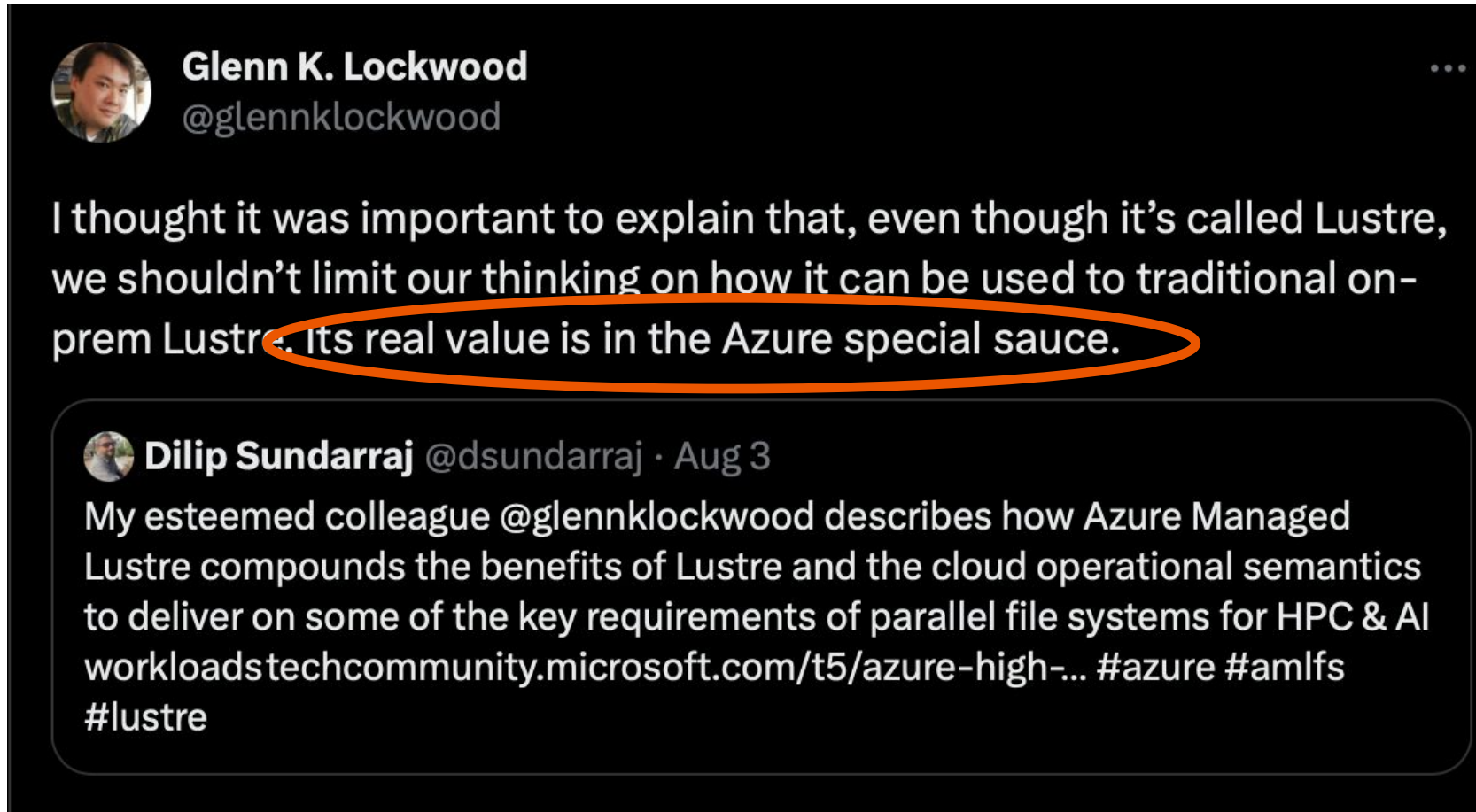
Automation for configuration, deployment, testing of distributed storage

“I find a bug in version V. I have a reproducer R. Someone submits patch P.”

*“How can I simply spin up a small system
running version V, apply patch P, and test with R?”*

– Thomas Bertschinger, LANL

TESSI: Reproducing Cloud Proprietary Magic in OSS



The full article is a quick informative read: [Azure Managed Lustre](#)

Using TESSI

Config file describes desired system (e.g. Lustre)

TESSI automates

- software installation,
- resource deployment,
- storage configuration (mounting, etc)
- testing

TESSI can run on bare metal -or- TESSI can create virtual clusters

TESSI is modular: storage system × virtual network

Storage-system modules

Lustre

data servers, metadata servers, clients

BeeGFS

management / metadata / storage / clients

pNFS

parallel NFS experiments and regression/debug workflows

Network modules

Dedicated secondary

best performance; requires physical network support

WireGuard

encrypted overlay over default network

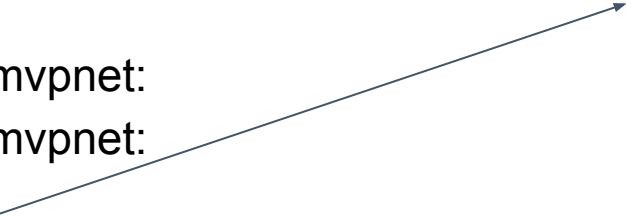
MVPNet

MPI-carried VM network; no host superuser privileges

TESSI in virtual mode using MVPNet allows automated storage deployment and testing WITHOUT root privilege!!!

Example TESSI Config File

```
all:
  vars:
    network:
      type: 'none'
    output_dir: "/proj/mvpnet/{{ current_user }}/experiments"
    system:
      type: 'pnfs'
      pnfs_src: '/proj/mvpnet/pnfs-lattice/'
      rondb_url: 'https://repo.hops.works/master/rondb-24.10.0-linux-.tar.gz'
    UNIQUE_TAG: "pmanual"
  children:
    physical_hosts:
      children:
        mds:
          hosts:
            tessi0.ubuntu.mvpnet:
            tessi3.ubuntu.mvpnet:
        ds:
          hosts:
            tessi1.ubuntu.mvpnet:
            tessi6.ubuntu.mvpnet:
          clients:
            hosts:
              tessi2.ubuntu.mvpnet:
              tessi9.ubuntu.mvpnet:
```



How Does TESSI Work in Virtual Mode?

It consumes a config file and creates a Makefile

The Makefile then executes multiple phases:

- 1. Gold VM images
 - 2. Virtual networking
 - 3. Cloning VMs
 - 4. Test networking
 - 5. Configure storage (mount, etc)
 - 6. Test the storage system
- 
- Executes on physical hosts
- Executes within virtual machines

How Does TESSI Work in Physical Mode?

It consumes a config file and creates a Makefile

The Makefile then executes multiple phases:

- ~~1. Gold VM images~~
- ~~2. Virtual networking~~
- ~~3. Cloning VMs~~
- ~~4. Test networking~~
5. Install required software
6. Configure storage (mount, etc)
7. Test the storage system

} Executes on physical hosts

dThreads:

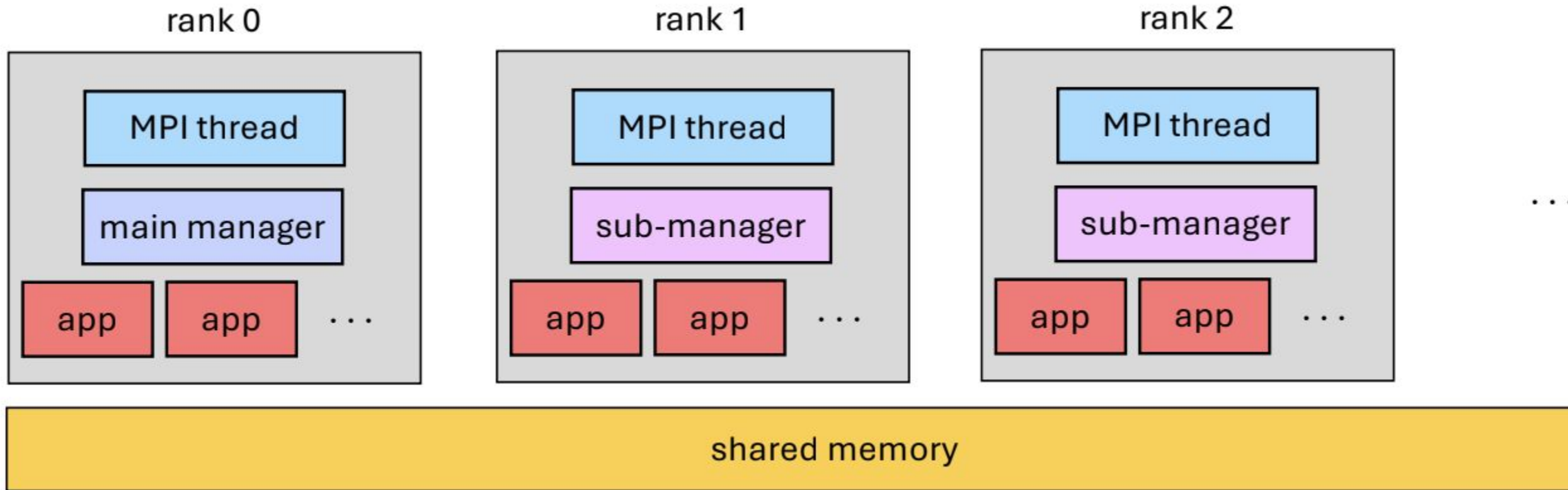
Using MPI to extend pthreads across multiple nodes

- Threads are fantastic programming model for single-node problems
- MPI is a fantastic programming model for large problems
 - If they lend themselves to message passing
- Transparently use MPI to spread threads across multiple nodes?
- Exemplar use cases
 - Massive graph exploration using graph data stored in CXL attached shared memory
- Note: This is an HPC-unique need
 - Cloud problems are shardable

dThread API

dthread API Function	pthread Equivalent	Semantic Result
<pre>err = dthread_create(&dthread _t, dthread_attr_t, <u>startfn</u>, arg)</pre>	<code>pthread_create</code>	Spawns a distributed thread using a dthreads-aware call, using MPI to ensure consistent state across physical nodes.
<pre>err = dthread_detach(dthread_ t)</pre>	<code>pthread_detach</code>	Detaches the thread to allow system resources to be reclaimed upon termination.
<pre>dthread_exit(retval)</pre>	<code>pthread_exit</code>	Terminates the calling <u>dthread</u> .
<pre>err = dthread_join(dthread_t, &retval)</pre>	<code>pthread_join</code>	Waits for the thread to complete execution and joins back to the main process.
<pre>err = dthread_cancel(dthread_ t)</pre>	<code>pthread_cancel</code>	Requests the cancellation of the specified <u>dthread</u> .

dThread Implementation



Summary

- MVPNet
 - Leveraging MPI to create virtual clusters that don't require root
- TESSI
 - Automated deployment and testing of distributed storage systems
- dThreads
 - Leveraging MPI to distribute threads across multiple physical nodes

Summary

- MVPNet
 - Leveraging MPI to create virtual clusters that don't require root
- TESSI
 - Automated deployment and testing of distributed storage systems
- dThreads
 - Leveraging MPI to distribute threads across multiple physical nodes



Shameless Plug:

- Follow my comedy account on Instagram
- https://www.instagram.com/john__b73/