ESnet
ENERGY SCIENCES NETWORK

Janus: Lightweight Container Orchestration for High-performance Data Sharing

Ezra Kissel
SNTA'22
June 30th 2022

U.S. DEPARTMENT OF ENERGY
Office of Science

BERKELEY LAB

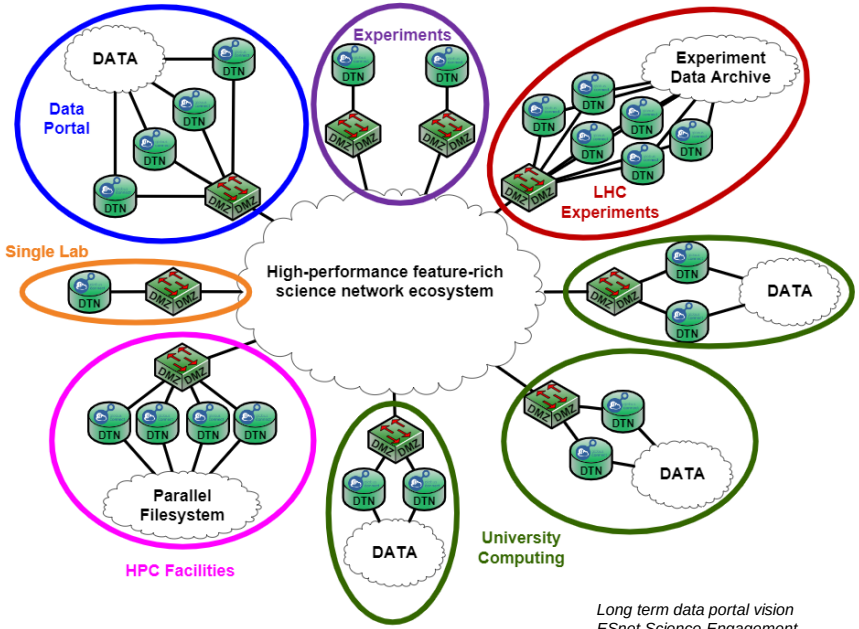
Overview and motivation

- Develop a managed data movement service capability
  - Support a pool of transfer software images that “just work”
  - Reduce reliance on varying levels of network/system expertise for deployments
  - Enable automated operations on Data Transfer Nodes (DTNs)
- Make use of containerization supported by lightweight orchestration
- Target high-speed data transfer deployments with dual-stack and multi-homed networking requirements
- Evaluate container networking with data transfer tools used in R&E nets

ESnet

Data Transfer Nodes

- DTNs fulfil a key role in enabling high-performance data movement
- Deployment models well-understood
  - There exists a wealth of knowledge in building, tuning, troubleshooting and maintaining DTN hardware and software
- Goal: expand availability, explore new usage models, and meet future demands → DTN-as-a-Service

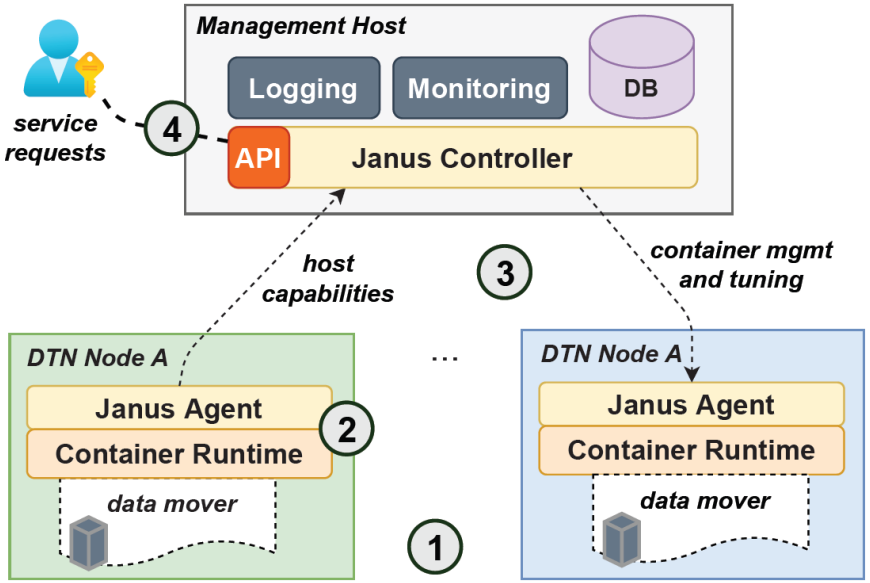


Long term data portal vision
ESnet Science Engagement

ESnet

Janus concept

1. Data mover software in containers
2. Network and storage performance optimization
3. Configuration and tuning flexibility
4. Lightweight service orchestration



ESnet

Extensible profiles

- Provide common configuration sets for service containers
- Helpful for consistency and re-use for larger deployments
- Specify capabilities once, then apply often

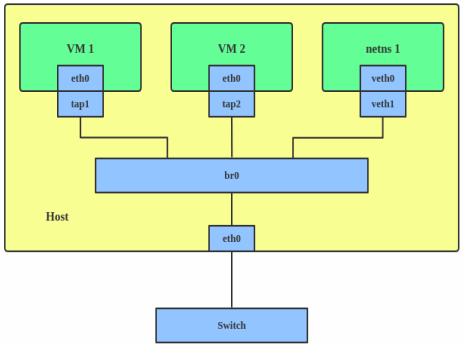
```
ID : Status              | Nodes/Services           | Image                                                | Profile
3  : STOPPED             | lbl-dev-dtn [None]       | wharf.es.net/dtnaas/opensciencegrid/cms-xcache:fresh | macvlan2x
5  : STARTED             | chic-cache1 [None]       | wharf.es.net/dtnaas/opensciencegrid/cms-xcache:fresh | chic-cms-xcache01
6  : STARTED             | bost-cache1 [None]       | wharf.es.net/dtnaas/opensciencegrid/cms-xcache:fresh | bost-cms-xcache01
janus>
janus> session create lbnl-tbn-1 image dtnaas/ofed profile lbnl-400g-1
```

```
features:
  rdmacap:
    devices:
      - devprefix: "/dev/infiniband"
        names:
          - rdma_cm
          - uverbs
    caps:
      - IPC_LOCK
      - NET_ADMIN
    limits:
      - Name: memlock
        Soft: -1
        Hard: -1

profiles:
  lbnl-400g-1:
    cpu: 4
    affinity: network
    mgmt_net: bridge
    data_net:
      name: net3001_eth200
      ipv4_addr:
        - 10.33.1.20
      ipv6_addr:
        - 2001:400:2202:2191::3
    features:
      - rdmacap
    privileged: false

  lbnl-400g-2:
    cpu: 8
    affinity: network
    mgmt_net: bridge
    data_net:
      name: net3002_eth200
      ipv4_addr:
        - 10.33.2.20
        - 10.33.2.21
    features:
      - rdmacap
    privileged: false
    volumes:
      - data
```

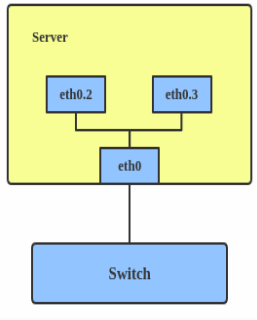
ESnet

Linux virtual networking and containers

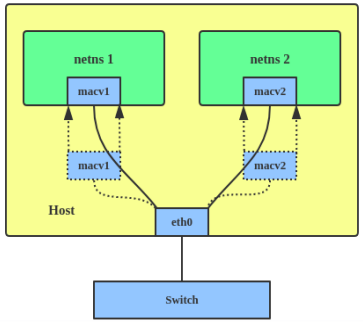
bridge[1]



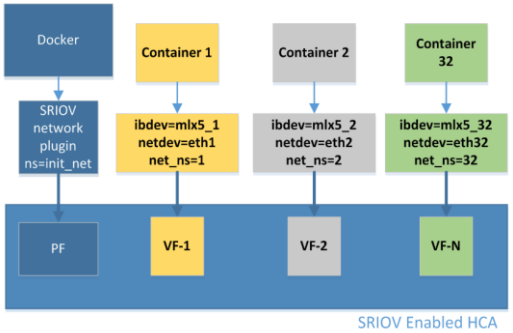
vlan[1]



macvlan[1]



sriov[2]



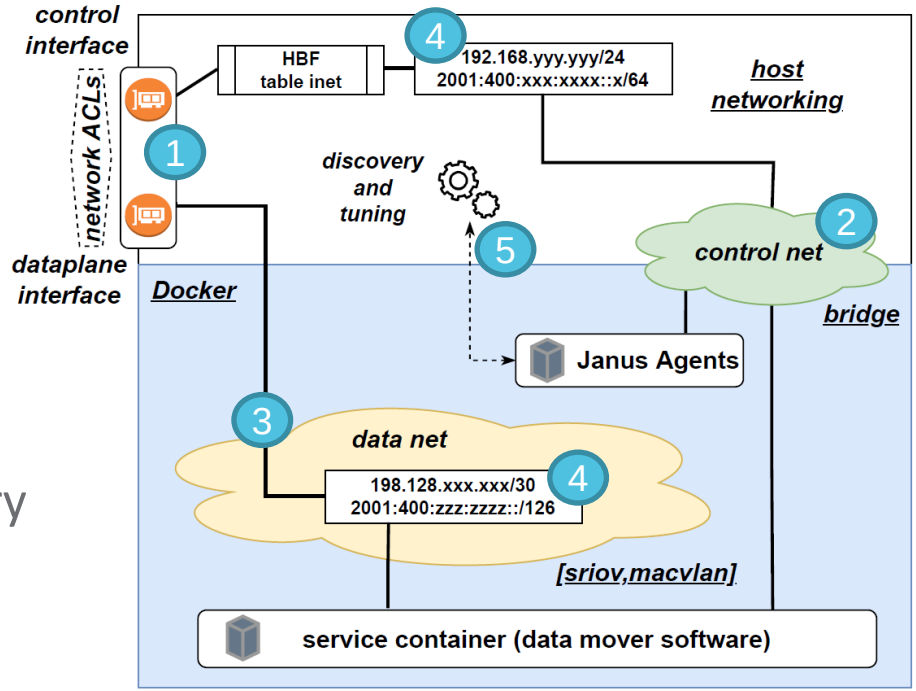
...and ipvlan, bonds, teams, etc.

Sources:
[1] https://developers.redhat.com/blog/2018/10/22/introduction-to-linux-interfaces-for-virtual-networking/
[2] https://community.mellanox.com/s/article/Docker-RDMA-SRIOV-Networking-with-ConnectX4-ConnectX5-ConnectX6

ESnet

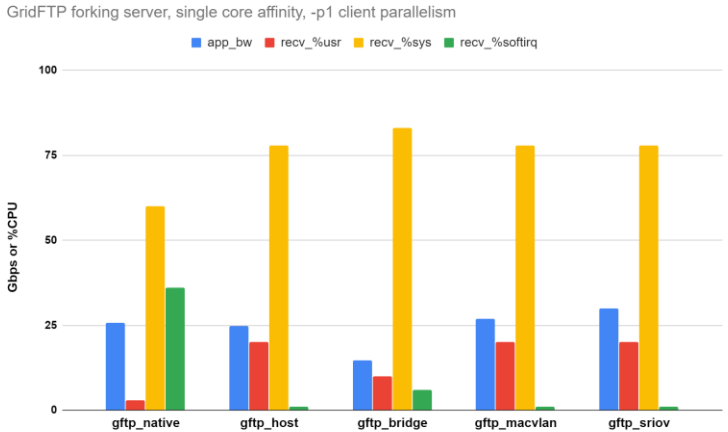
Example data transfer node configuration

1. Multi-homed physical nodes
2. Slow path control
3. Fast path data plane
4. Dual-stack IP networking
5. Local agents for resource discovery and customized tuning



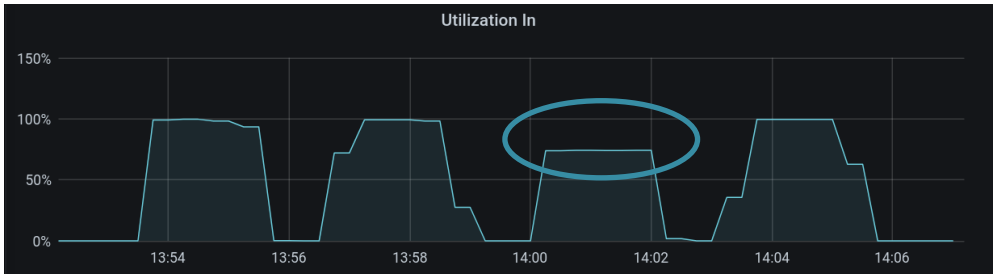
Container networking performance

- A number of container network attachments are supported and available
  - Which works best for high-performance networking in each deployment?
- Relatively easy to achieve matching performance using tuning params
  - E.g., container execution mapped to appropriate NUMA nodes for cores and memory

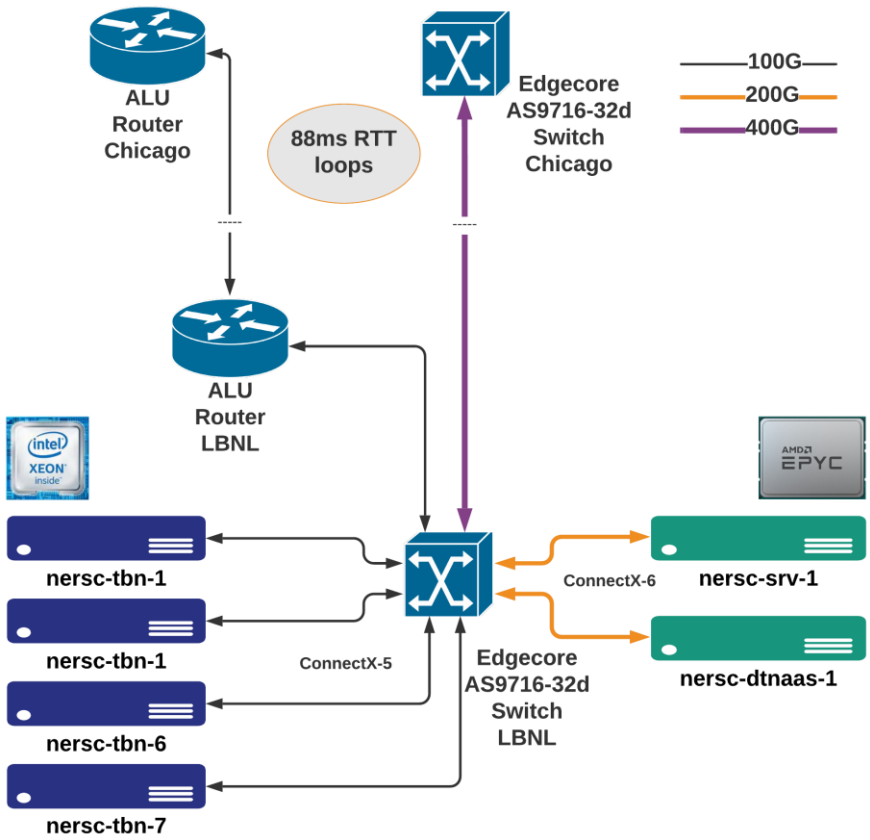


100G links, GridFTP, forking server, 5 clients with -p4 parallelism

native system *container host net* *container bridge net* *container macvlan/sriov net*

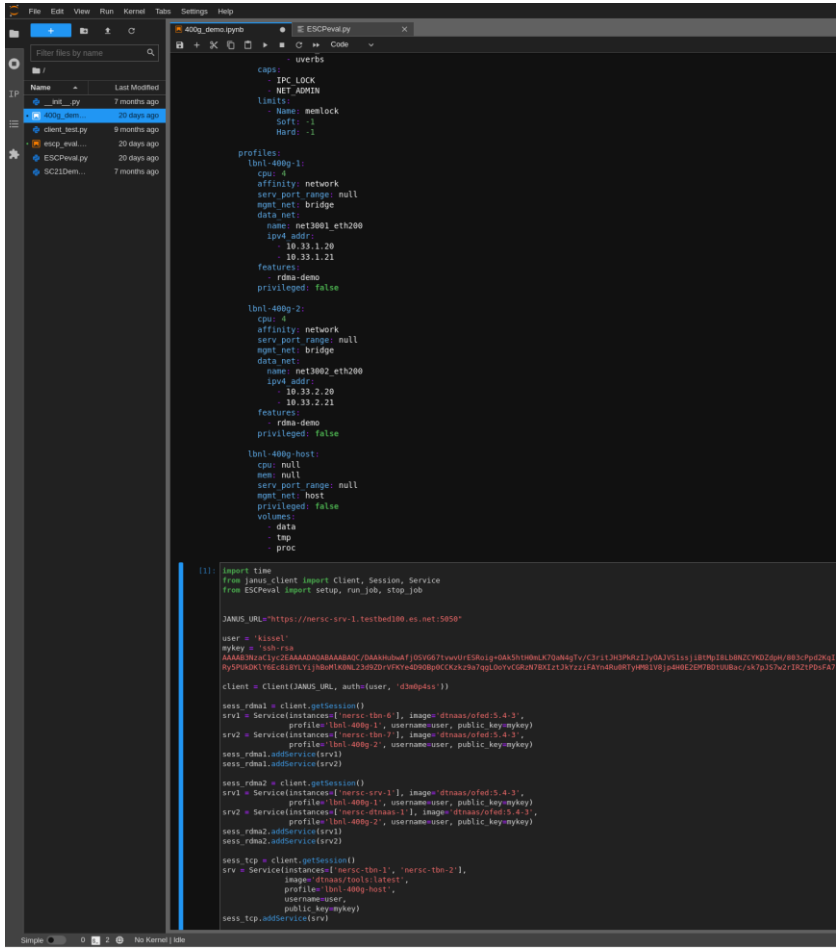


Evaluation topology (ESnet testbed)

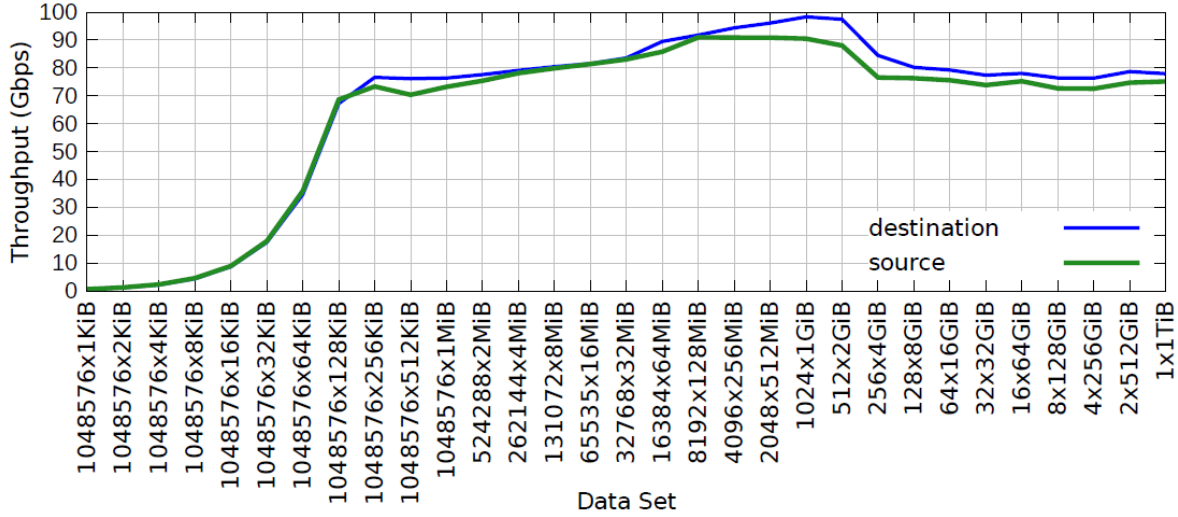


Evaluation environment

- Performant data movers: EScp, GridFTP, mdtmFTP, Zettar zx
- Janus Python client and CLI
  - Programmatic execution
  - Jupyter Notebook scripting
- Standard 100G host and network interface tuning



Storage baseline

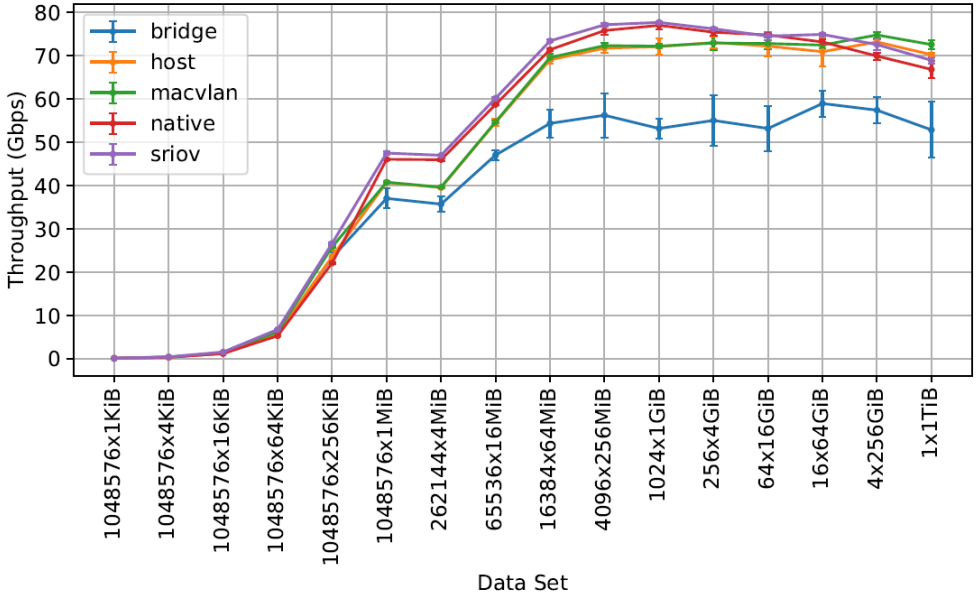


elbencho storage sweep, sequential write benchmark

https://github.com/breuner/elbencho/tree/master/contrib/storage_sweep

ESnet

Container network comparison (LAN)

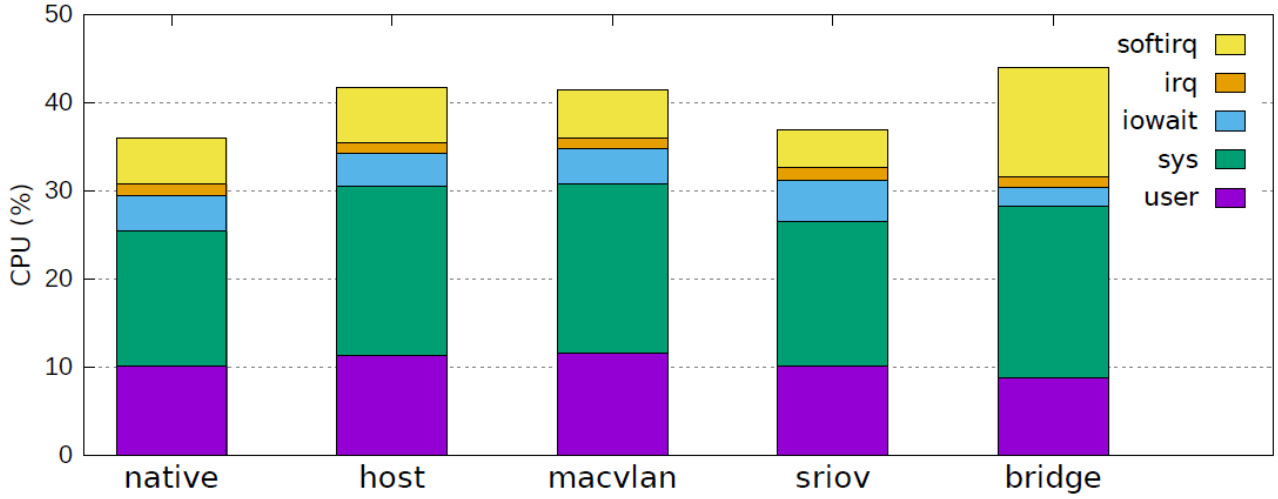


Each src/dst container instance launched by Janus with specified net profile

SR-IOV provides best performance compared to native

ESnet

CPU usage comparison

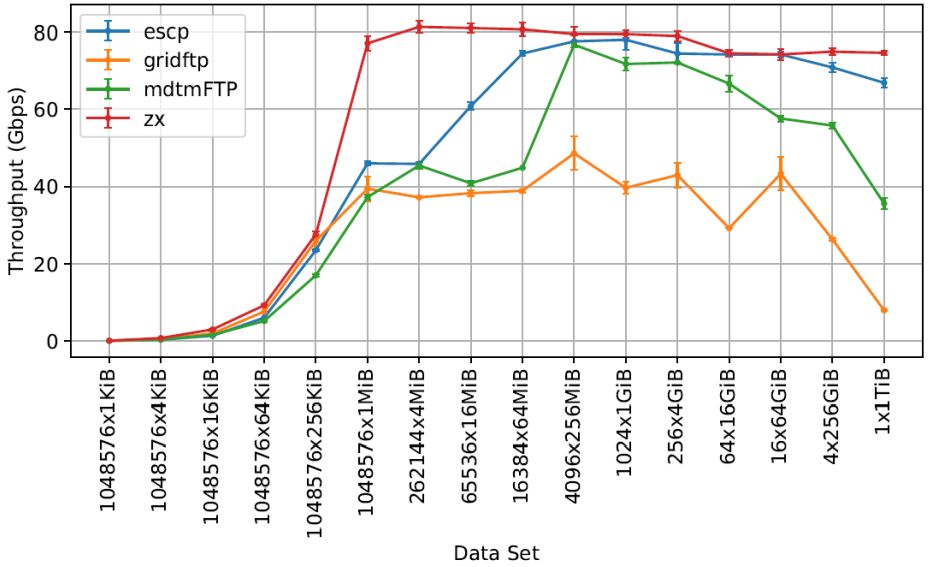


CPU overheads are a non-trivial factor, destination usage shown

Bridge networking incurs the highest overheads

ESnet

Data mover comparison over WAN



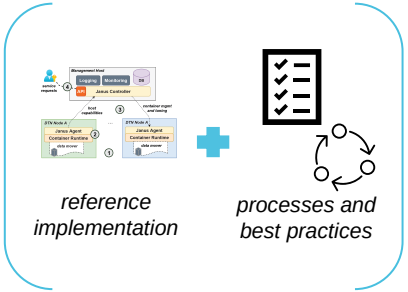
Performance of various transfer software over 88ms WAN link

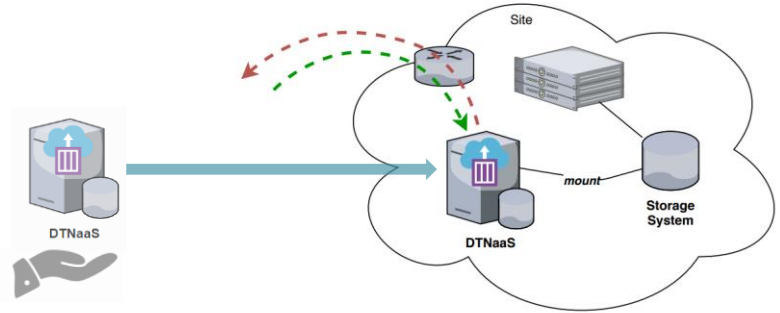
Transfer parallelism approach has a large impact across file size distributions

ESnet

Conclusion and Future work

- Containers for high performance data movement is a viable approach
- 200G/400G evaluation
- Extend tuning automation
- Provide guidance for best practice and common concerns
  - Performance, security, network orchestration, measurement and monitoring





ESnet

Thank you

Questions?

Ezra Kissel

kissel@es.net

ESnet