

A Result-Data Offloading Service for HPC Centers

Henry Monti and Ali R. Butt
{hmonti, butta} @cs.vt.edu
<http://research.cs.vt.edu/dssl/>
Distributed Systems and Storage Laboratory



Sudharshan S. Vazhkudai
vazhkudaiss@ornl.gov
<http://www.csm.ornl.gov/~vazhkuda/Storage.html>
Oak Ridge National Laboratory



HPC Center Data Offload Problem

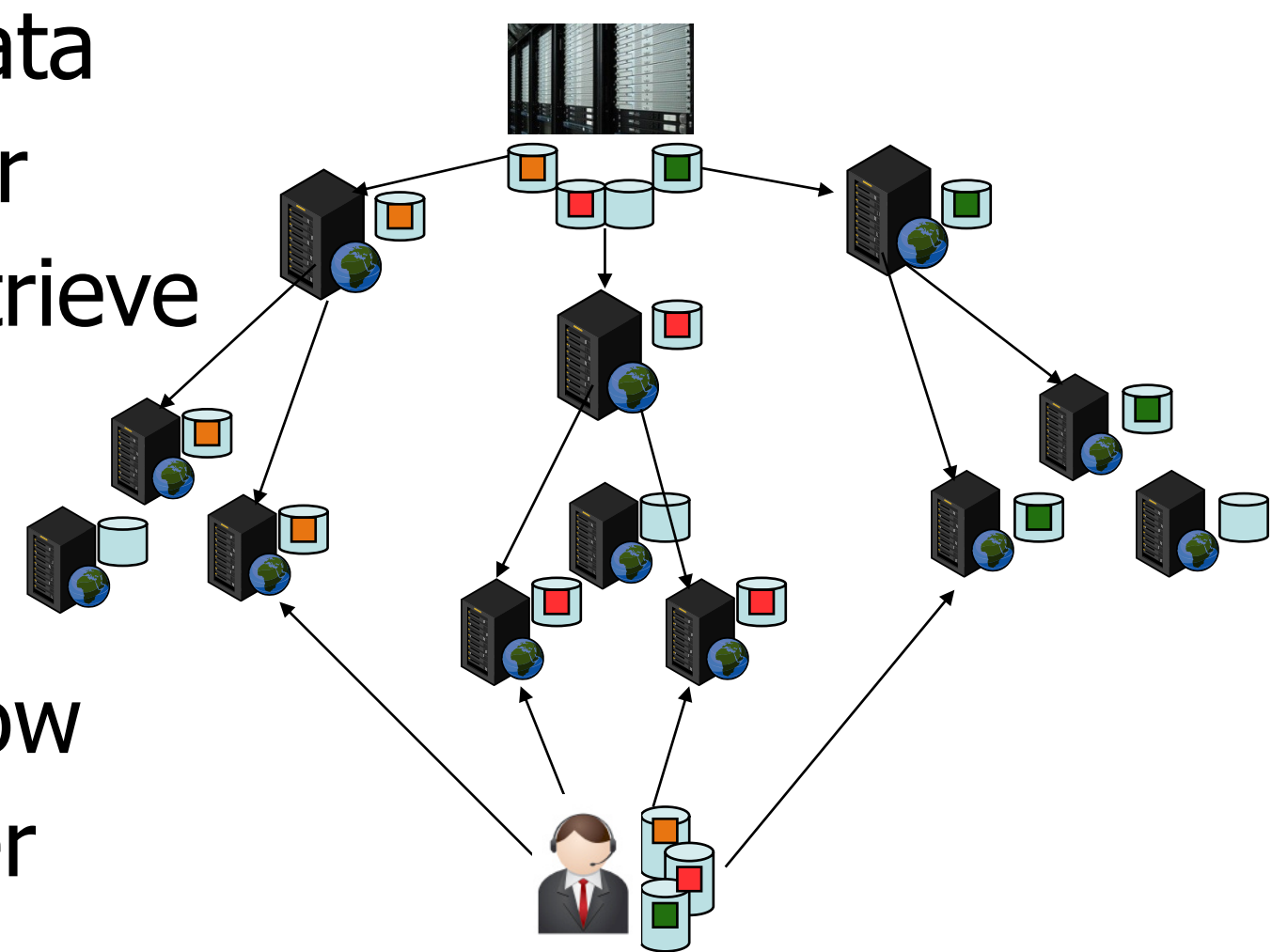
- Offloading errors affect Supercomputer serviceability
 - Offloading is a large data job prone to failure
 - End resource unavailability, Transfer errors
- Delayed offloading
 - From a center standpoint:
 - Wastes scratch space
 - Renders result data vulnerable to purging
 - From a user job standpoint:
 - Increased turnaround time if part of the job workflow depends on offloaded data
 - Potential resubmits due to purging

Limitations of Current Offloading

- Home-grown solutions
- Utilize point-to-point direct transfers
 - GridFTP, HIS, scp
- Require end resources to be available
- Do not exploit orthogonal bandwidth
- Do not consider SLAs or purge times
- **Direct transfers not ideal for data offloading**

A Decentralized Offloading Service

- Utilize army of intermediary storage locations
- Offload data to nearby nodes
- Support multi-hop data migration to end user
- Allow end user to retrieve data as necessary
- Provide multiple fault-tolerant data flow paths from the center to the user



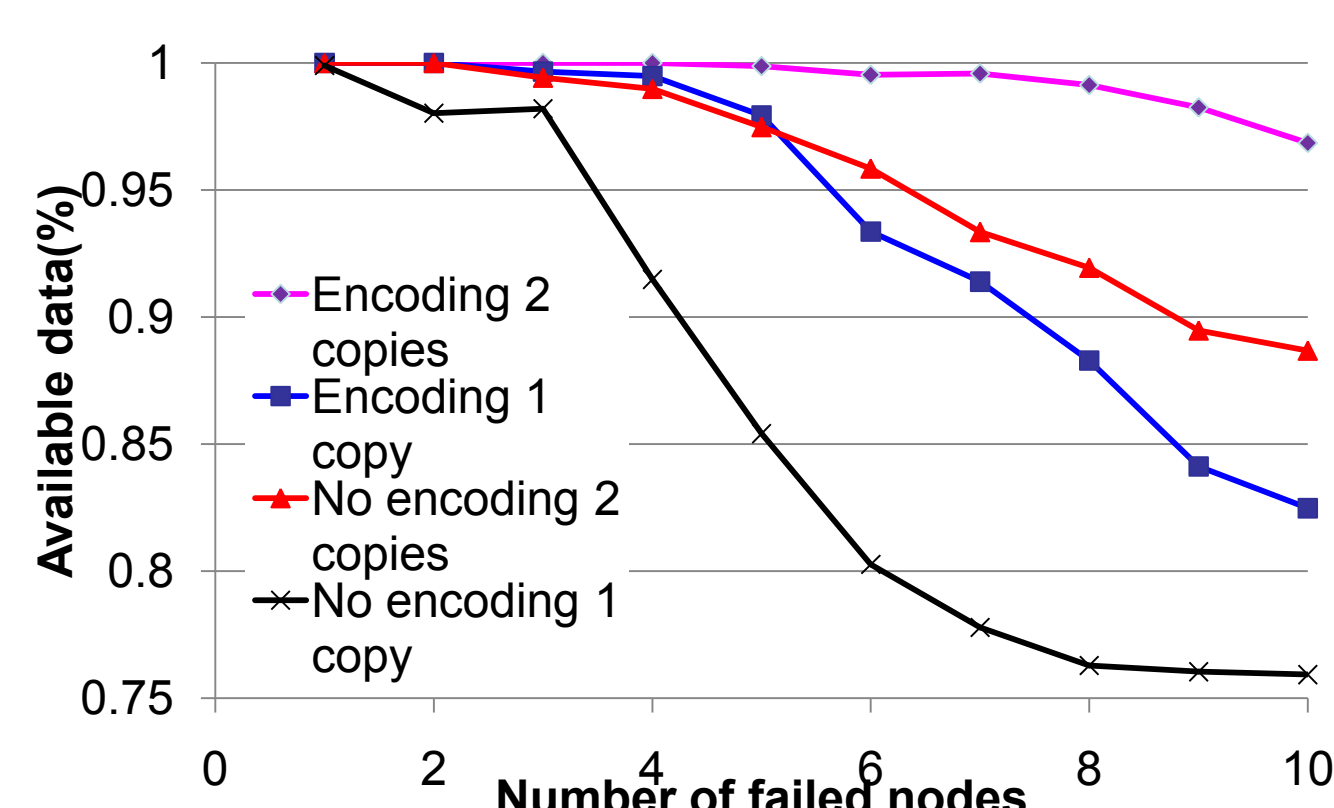
Challenges Faced in Our Design

- Discovering intermediary nodes
 - Use structured peer-to-peer networks
- Addressing insufficient participants
 - Utilize Landmark nodes
- Adapting to dynamic network behavior
 - Employ bandwidth monitoring
- Ensuring data reliability and availability
 - Utilize replication and erasure-coding

Evaluation using PlanetLab

- 22 PlanetLab nodes center + end user + 20 intermediary nodes
- Compare direct with the proposed method
 1. Random distribution
 2. Bandwidth measurement based
 3. Bandwidth forecasts based
- Study Replication vs. Erasure coding

	Direct	Rand	Measmt. Based	Forecast Based
Offload	739	245	214	210
Push	N/A	431	393	370
Pull	739	665	663	663



Conclusion

- A fresh look at Offloading
 - Decentralized approach
 - Monitoring-based adaptation
- Considers SLAs and purge policies
- Provides high reliability for data
- Outperforms direct transfer by **72%**
- Open issues:
 - Strategically placed Landmark nodes
 - Eager offloading
 - Integration with job script