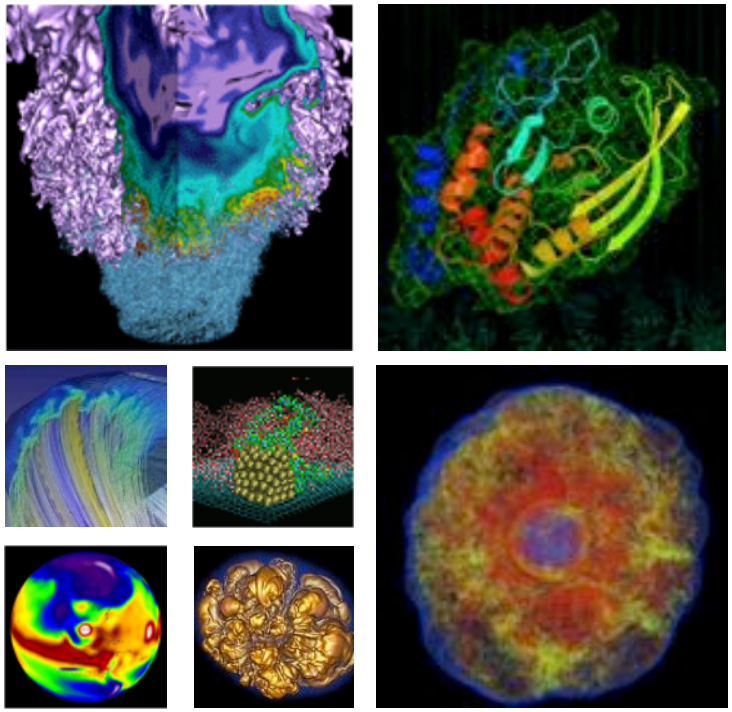
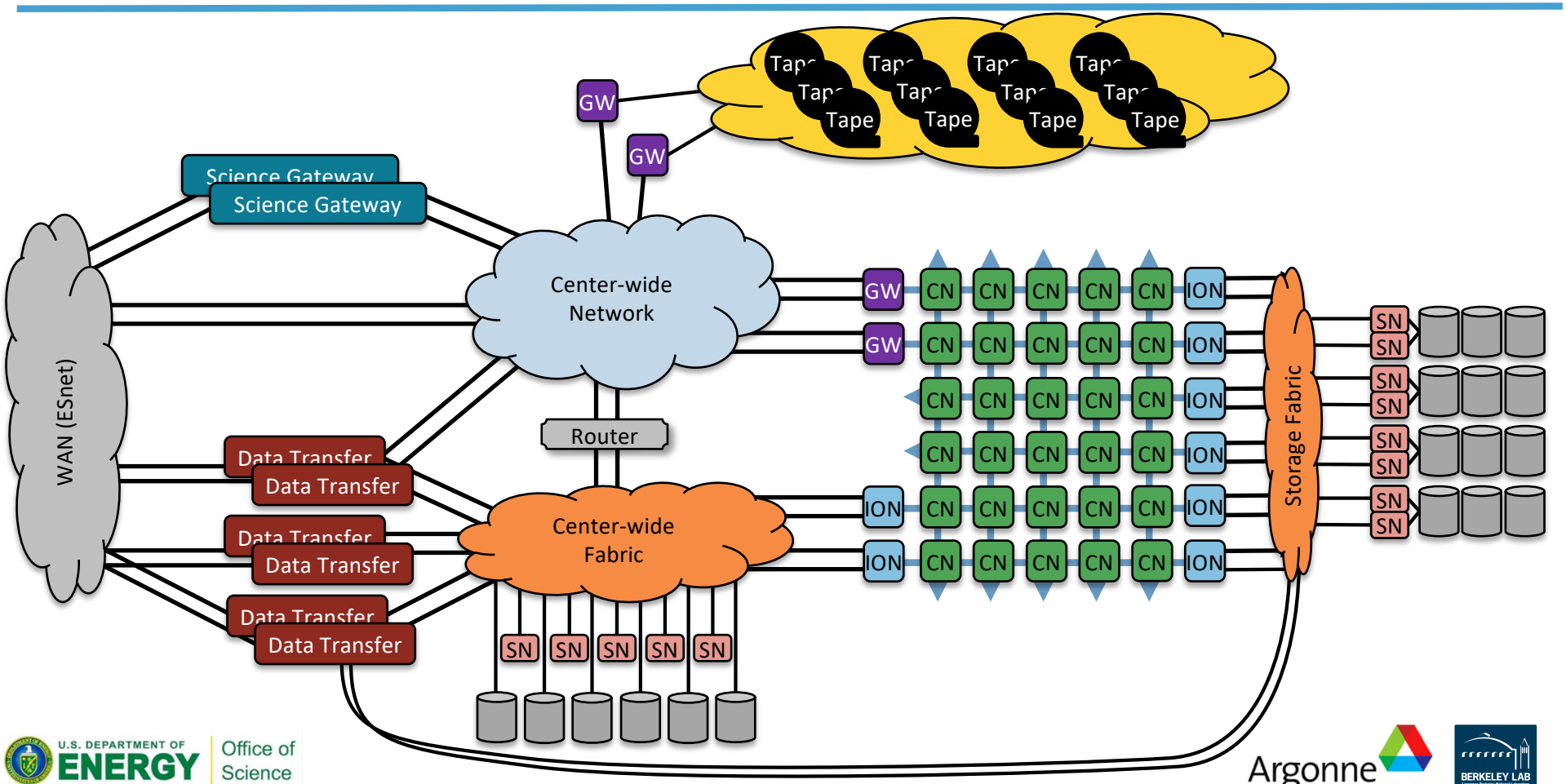
Understanding Data Motion in the Modern HPC Data Center



Glenn K. Lockwood
Shane Snyder
Suren Byna
Philip Carns
Nicholas J. Wright

Scientific computing is more than compute!

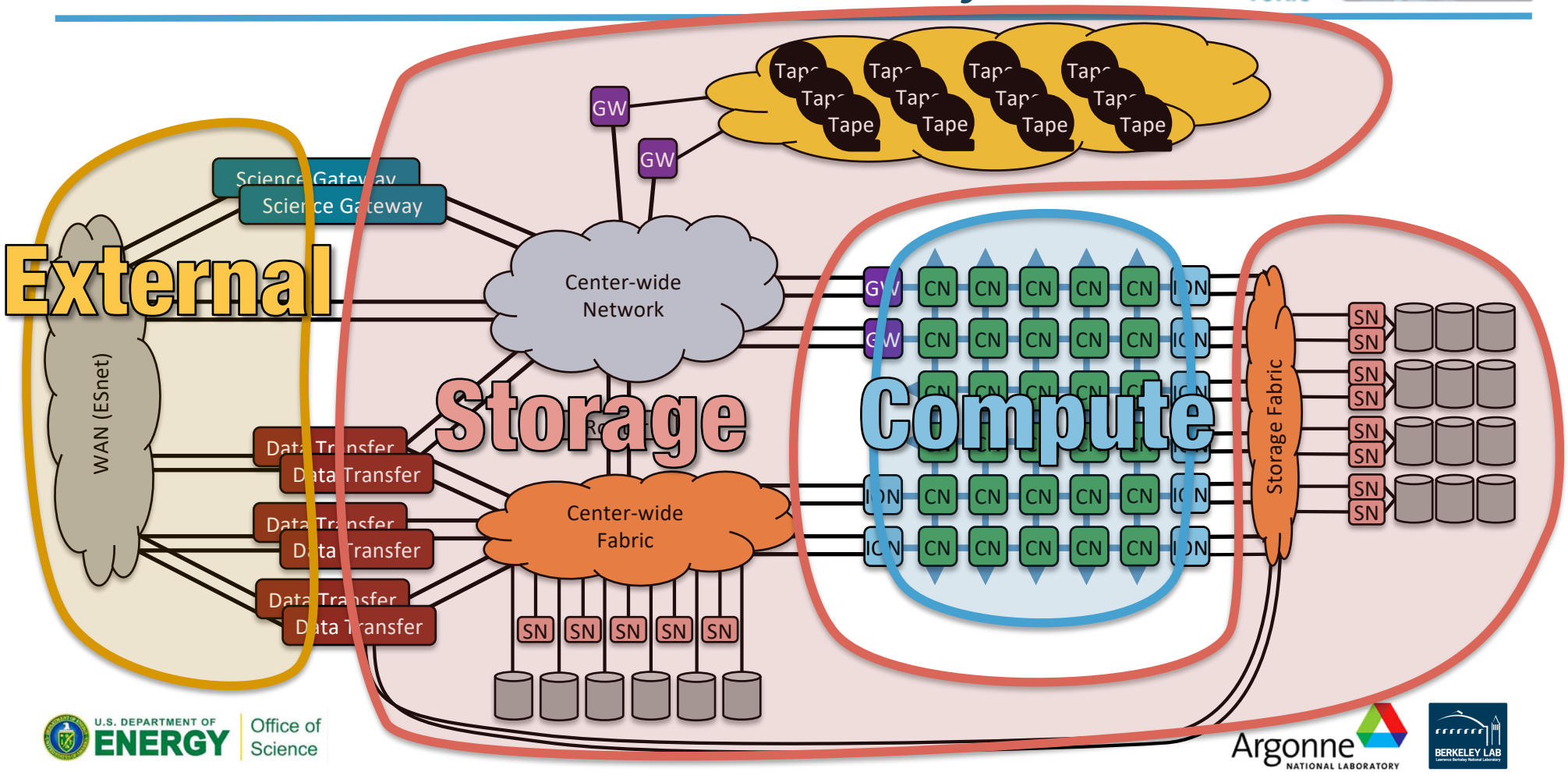


U.S. DEPARTMENT OF ENERGY | Office of Science

Argonne NATIONAL LABORATORY

BERKELEY LAB

Goal: Understand data motion *everywhere*

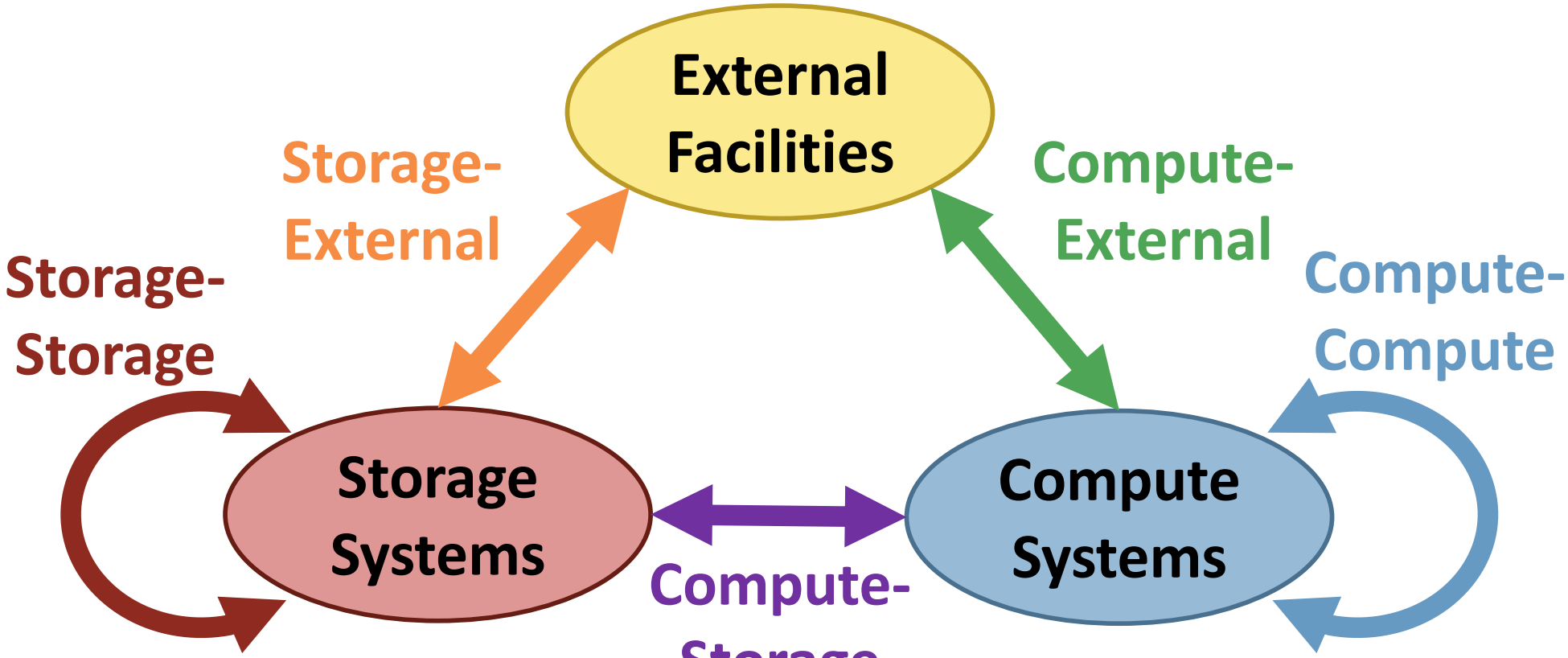


U.S. DEPARTMENT OF ENERGY | Office of Science

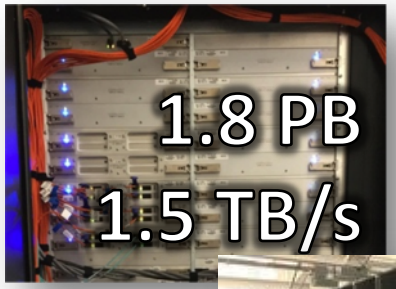
Argonne NATIONAL LABORATORY

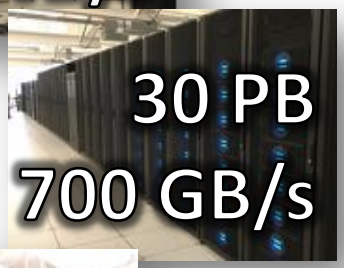
BERKELEY LAB

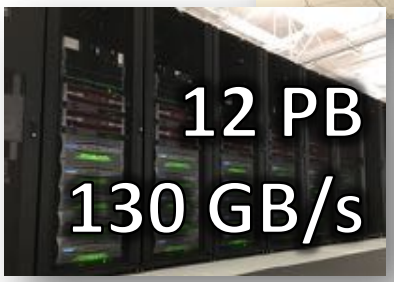
Our simplified model for data motion

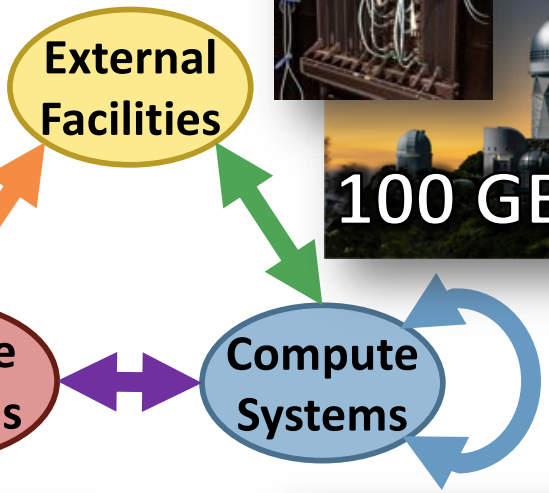


Mapping this model to NERSC







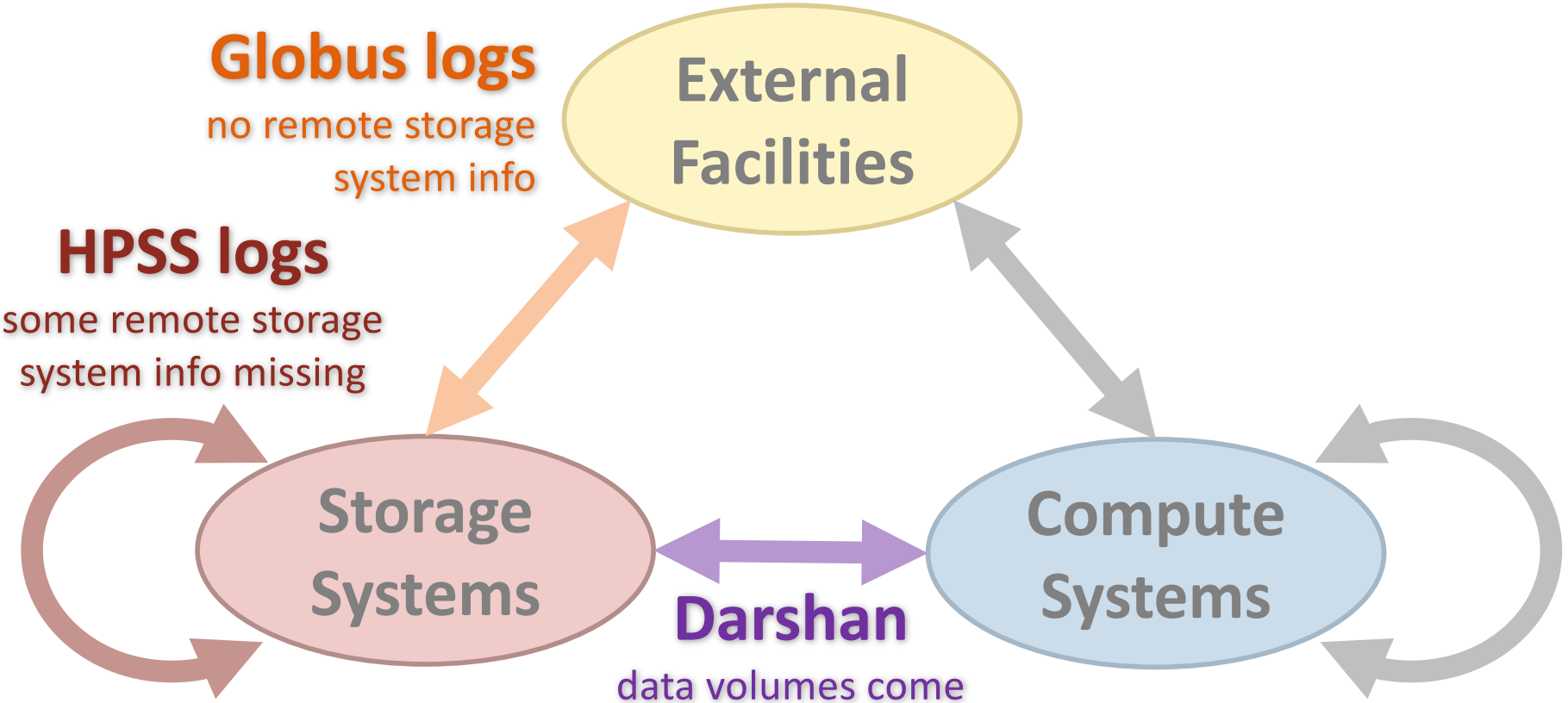


U.S. DEPARTMENT OF ENERGY | Office of Science

Argonne NATIONAL LABORATORY

BERKELEY LAB

Relevant logs kicking around at NERSC



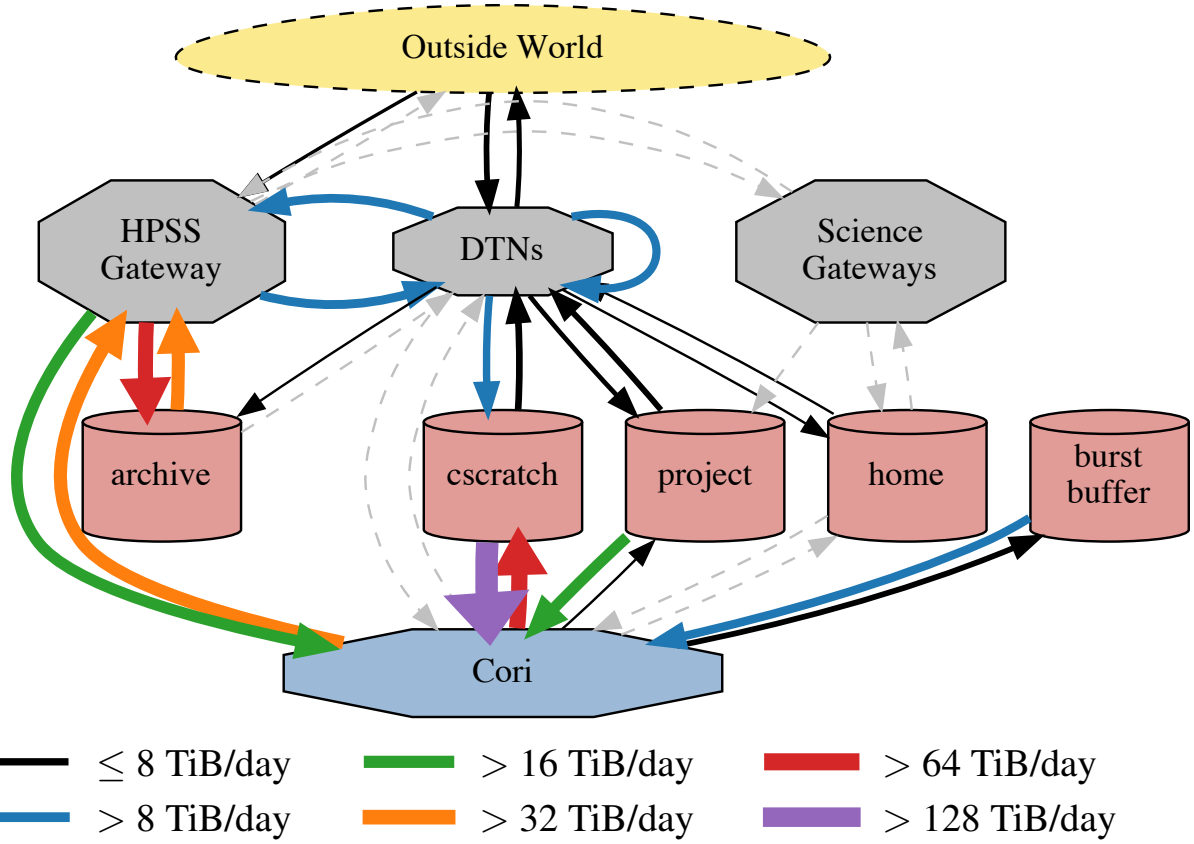
Normalizing data transfer records

Parameter	Example
Source site, host, storage system	NERSC, Cori, System Memory
Destination site, host, storage system	NERSC, Cori, cscratch1 (Lustre)
Time of transfer start and finish	June 4 @ 12:28 – June 4 @ 12:32
Volume of data transferred	34,359,738,368 bytes
Tool that logged transfer	Darshan, POSIX I/O module
Owner of data transferred	uname=glock, uid=69615

What is possible with this approach?

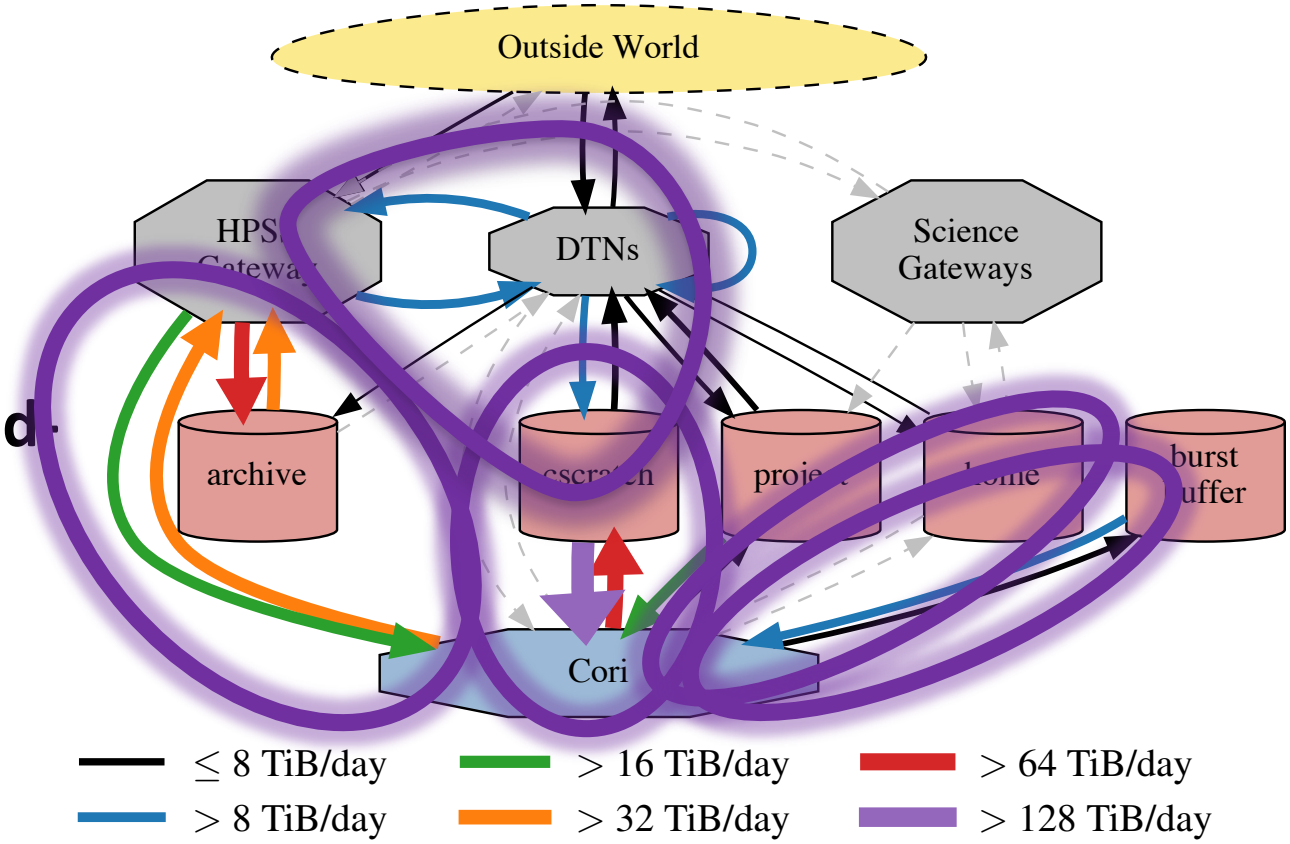
May 1 – August 1, 2019

- 194 million transfers
- 78.6 PiB data moved

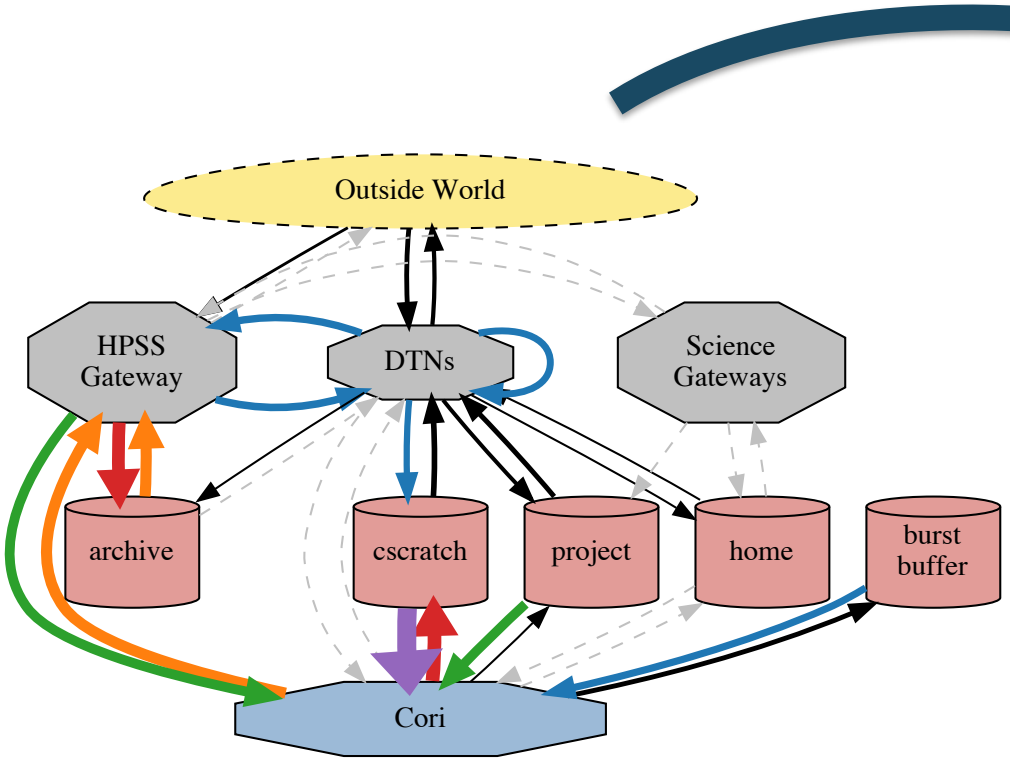


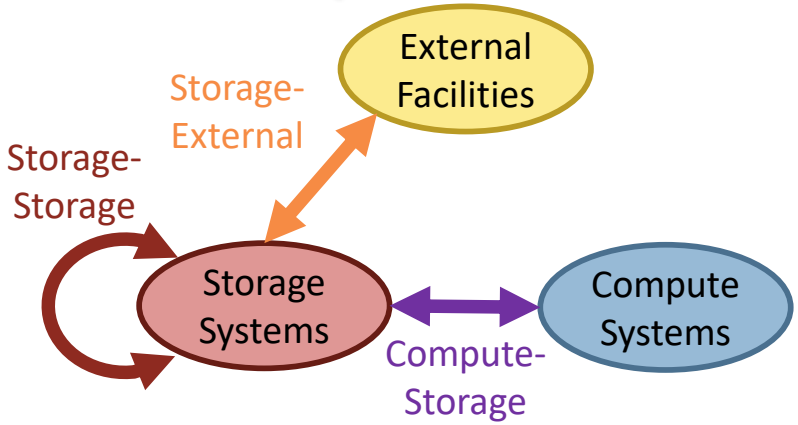
Visualizing data motion as a graph

- Job I/O is most voluminous
- Home file system usage is least voluminous
- Burst buffer is read-heavy
- Users prefer to access archive directly from Cori than use DTNs



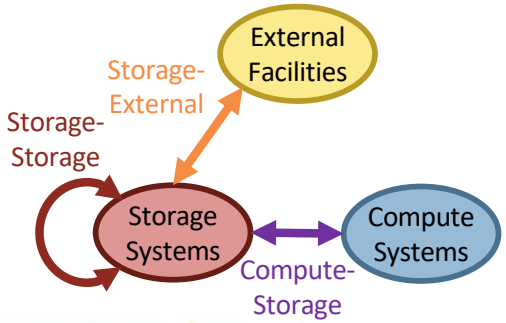
Mapping this data to our model

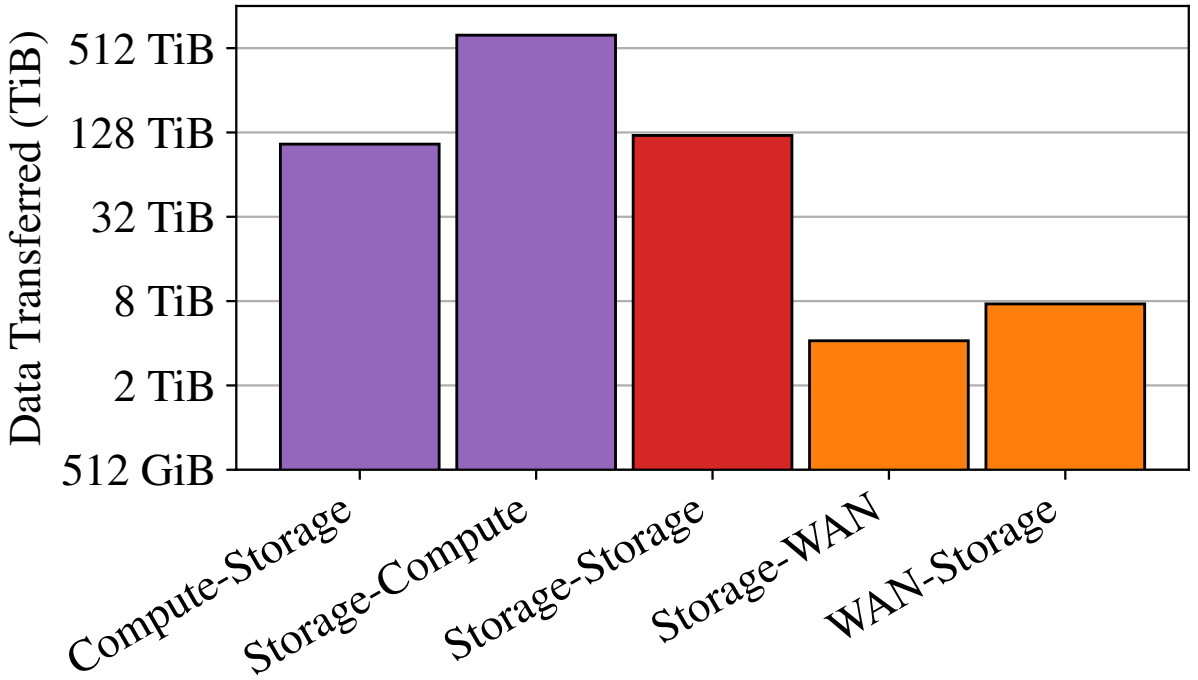




Adding up data moved along each vector

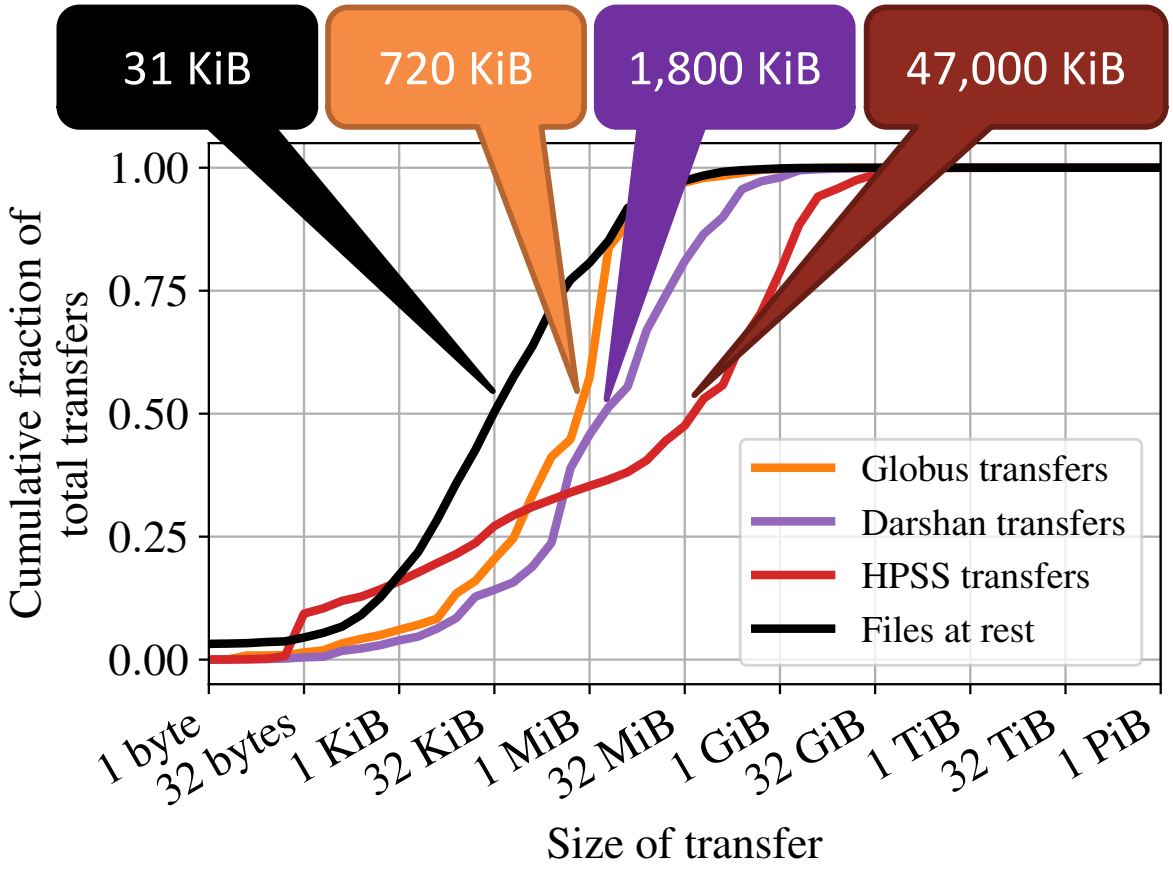
- **Job I/O is significant**
- **Inter-tier is significant**
  - I/O outside of jobs ~ job write traffic
  - Fewer tiers, fewer tears
- **HPC I/O is not just checkpoint-restart!**



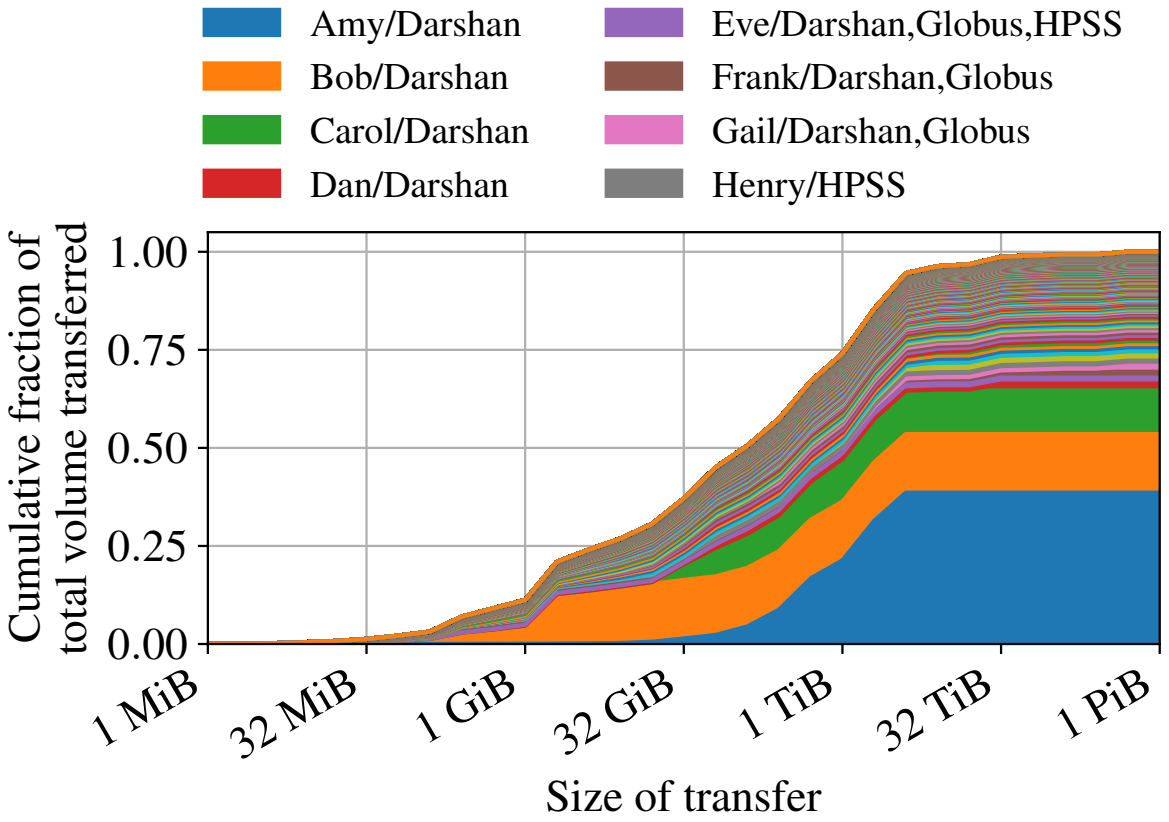


Examining non-job I/O patterns

- **Hypothesis: non-job I/O is poorly formed**
  - Job I/O: optimized
  - Others: fire-and-forget
- **Users transfer larger files than they store (good)**
- **Archive transfers are largest (good)**
- **WAN transfers are smaller than job I/O files (less good)**



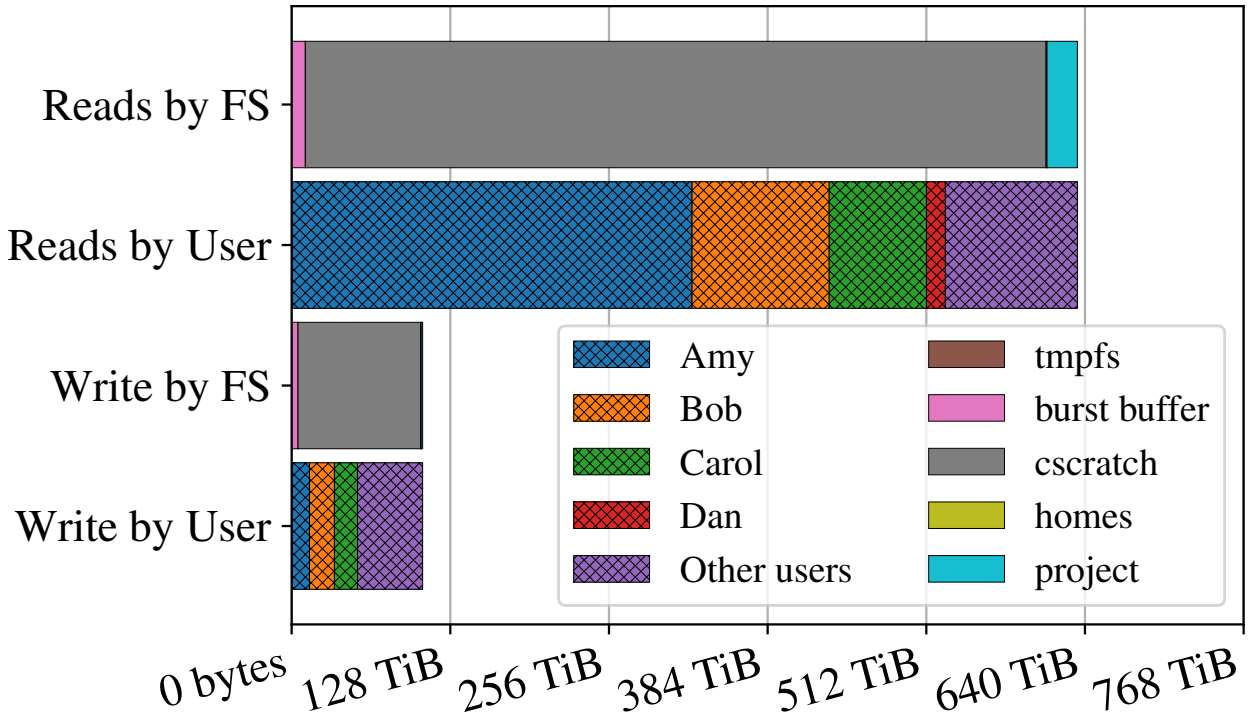
Few users resulted in the most transfers



- **1,562 unique users**
- **Top 4 users = 66% of volume transferred**
- **Users 5-8 = 5.8%**
  - All used multiple transfer vectors
  - Henry is a storage-only user

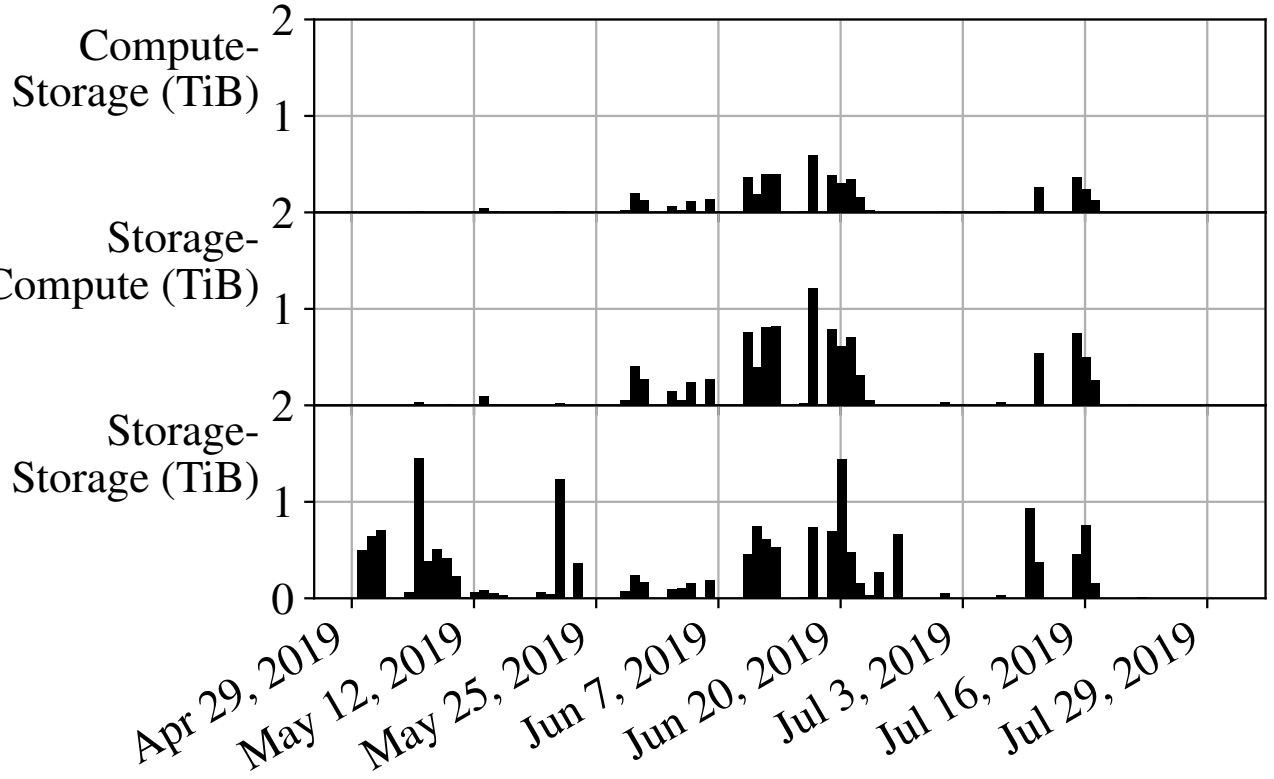
Examining transfers along many dimensions

- **Break down transfers by r/w and file system**
- **Top users are read-heavy**
  - Rereading same files
  - Targeting cscratch (Lustre)



Tracing using users, volumes, and directions

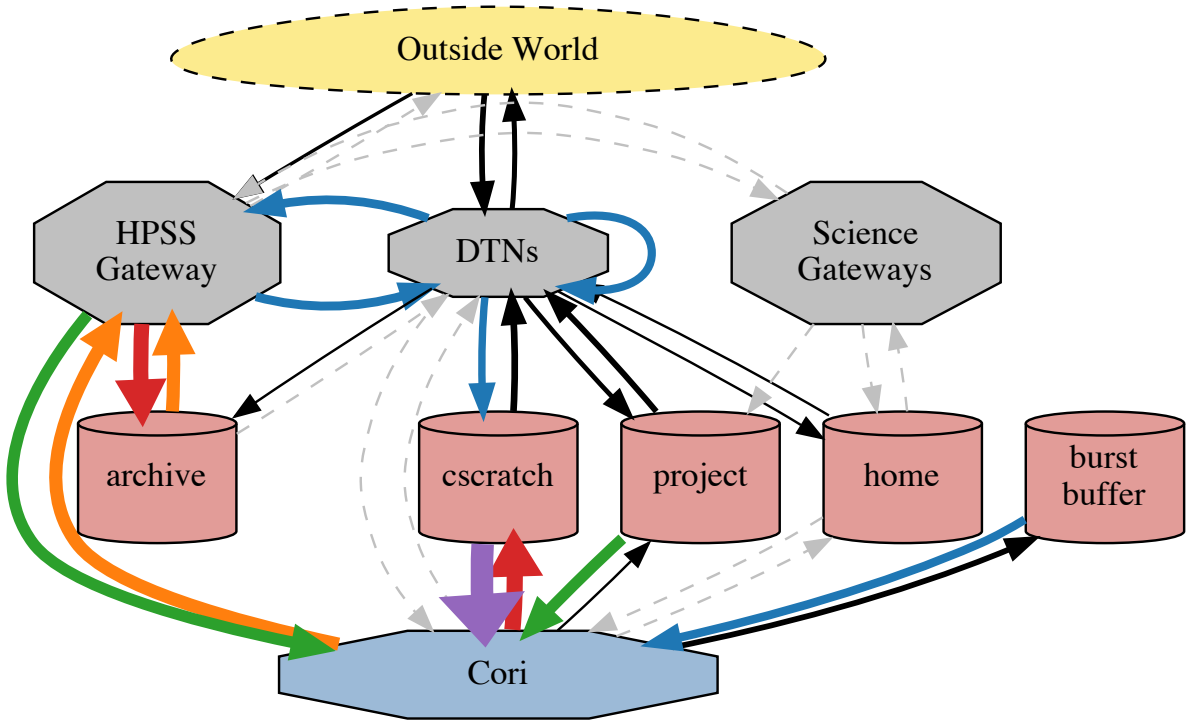
- **Correlating reveals workflow coupling**
  - S-S precedes C-S/S-C
  - 2:1 RW ratio during job
  - Data reduction of archived data
- **This was admittedly an exceptional case**



Is this the full story?

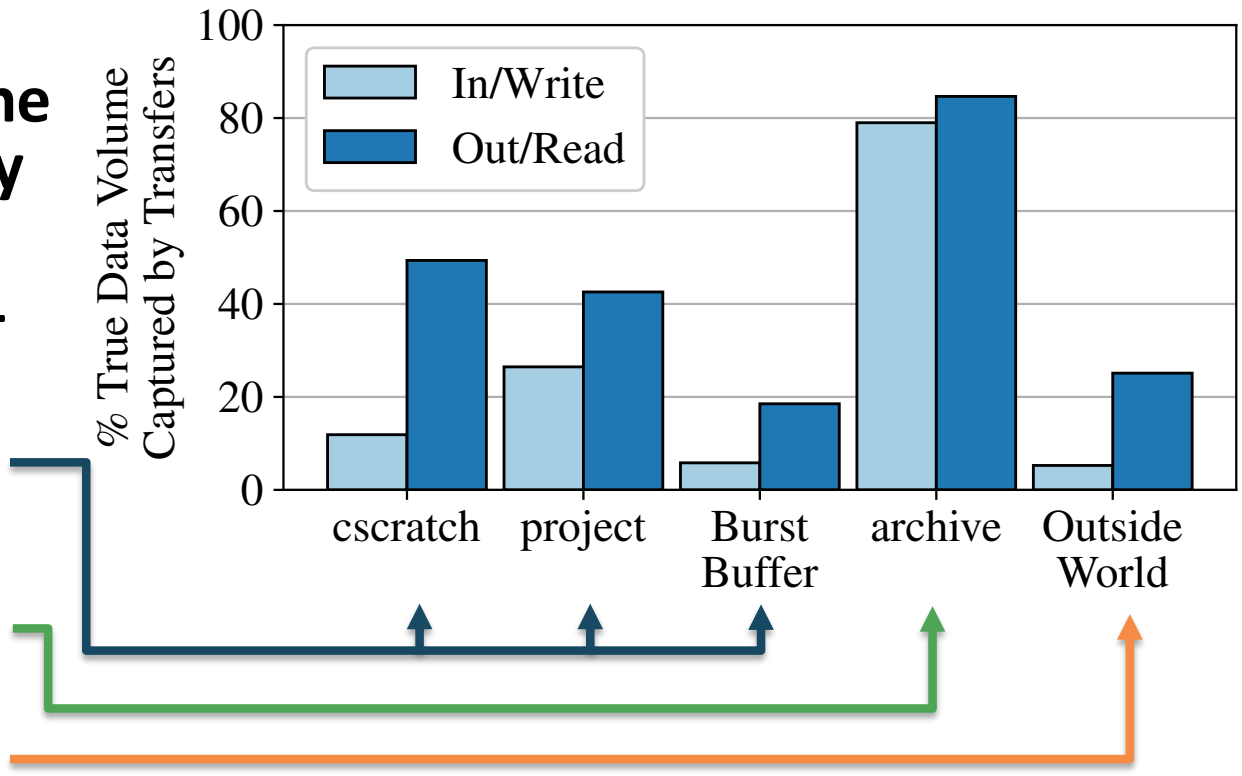
Quantify the amount of transfers *not* captured

- Compare volume transferred to system monitoring (storage systems)
- Compare bytes in to bytes out (transfer nodes)



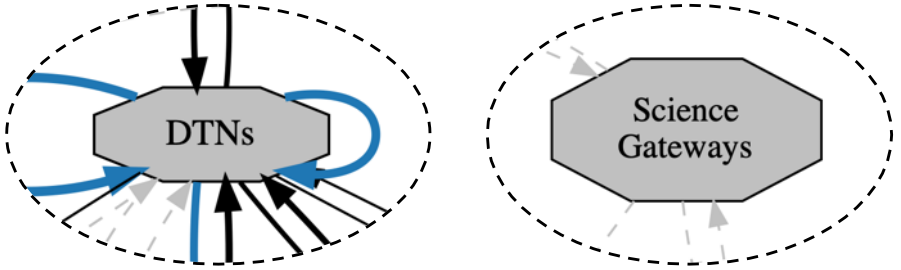
Not every data transfer was captured

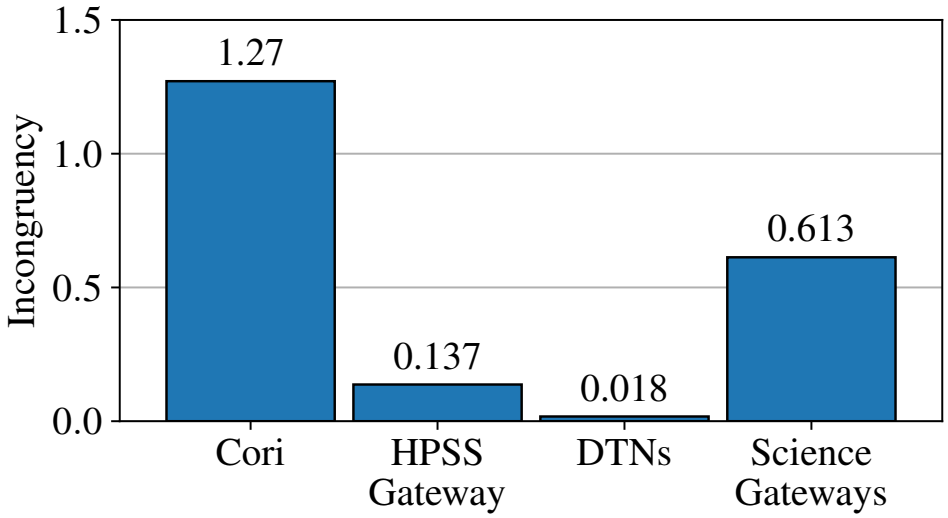
- **100% true data volume should be captured by transfers**
- **Missing lots of data—why?**
  - **Darshan logs not generated; cp missing**
  - **Globus-HPSS adapter logs absent**
  - **Only Globus logged; rsync/bbcp absent**



Identifying leaky transfer nodes

- **Incongruency (Δ)**
  - data in vs. data out
  - FOM for how "leaky" a node is
  - Δ = 0 means all bytes in = all bytes out
- **Cori: expect >> 0 because jobs generate data**
- **Science gateways > 0 because ???**



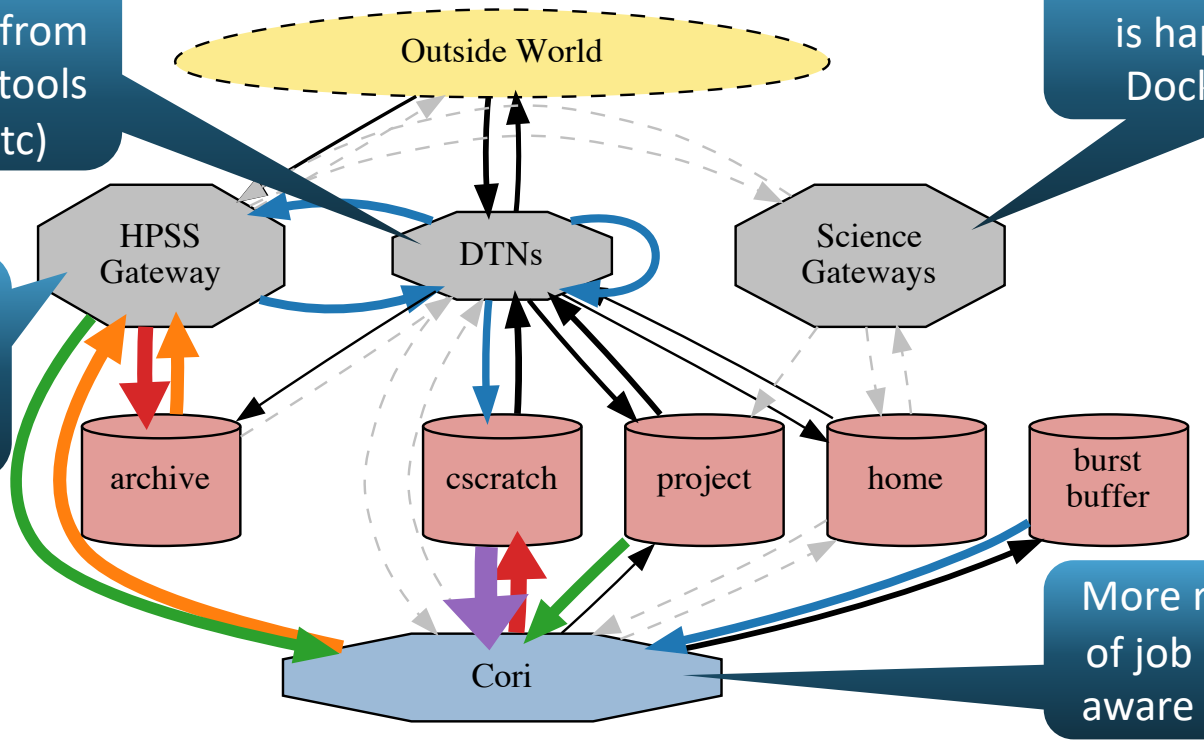


Towards Total Knowledge of I/O

New profiling tools to capture I/O from other transfer tools (bbcp, scp, etc)

Better insight into what is happening inside Docker containers

Improve analysis process to handle complex transfers



More robust collection of job I/O data; cache-aware I/O data (LDMS)

There's more to HPC I/O than job I/O

- **Inter-tier I/O is too significant to ignore**
  - need better monitoring of data transfer tools
  - users benefit from fewer tiers, strong connectivity between tiers
  - need to optimize non-job I/O patterns
- **Transfer-centric approaches yield new holistic insight into workflow I/O behavior**
  - Possible to trace user workflows across a center
  - Humans in the loop motivate more sophisticated methods

TOKIO

We gratefully acknowledge the support of

- Damian Hazen (NERSC)
- Kristy Kallback-Rose (NERSC)
- Nick Balthaser (NERSC)
- Lisa Gerhardt (NERSC)
- Ravi Cheema (NERSC)
- Jon Dugan (ESnet)
- Eli Dart (ESnet)

We're hiring!

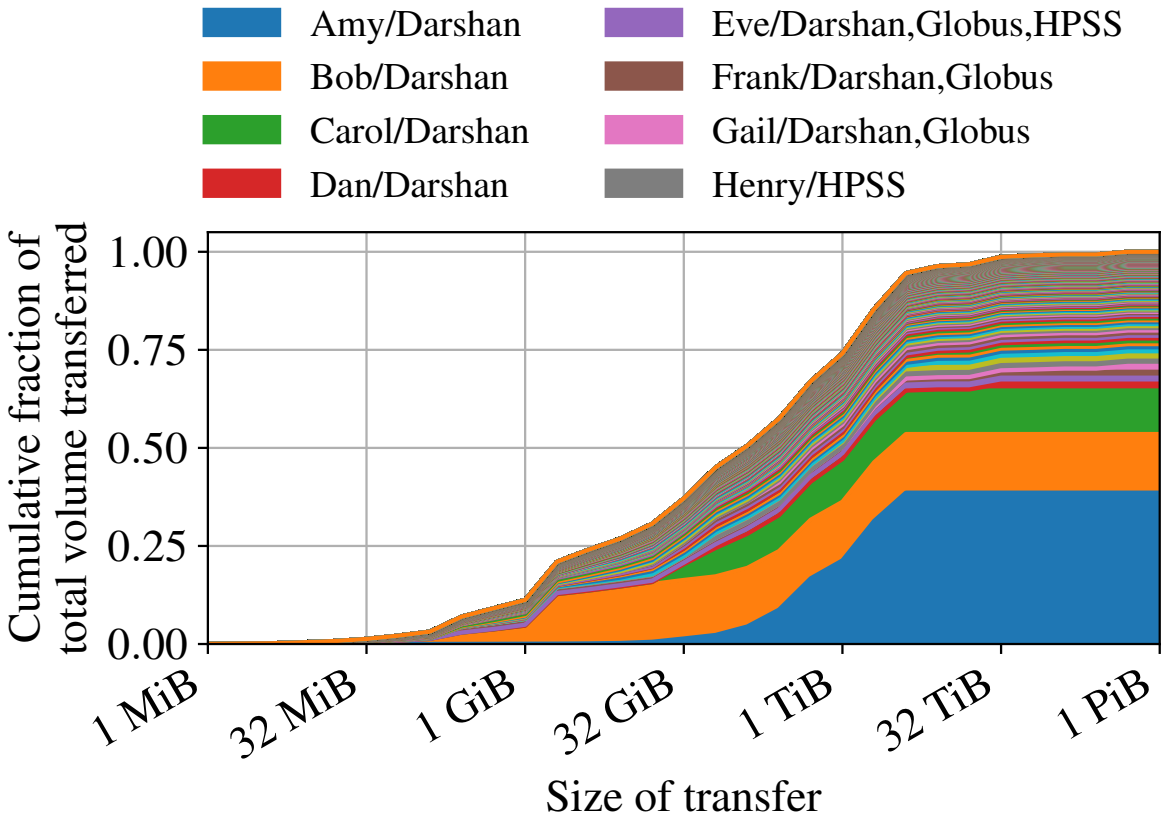
This material is based upon work supported by the U.S. Department of Energy, Office of Science, under contracts **DE- AC02-05CH11231** and **DE-AC02-06CH11357**. This research used resources and data generated from resources of the **National Energy Research Scientific Computing Center**, a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231 and the **Argonne Leadership Computing Facility**, a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357.

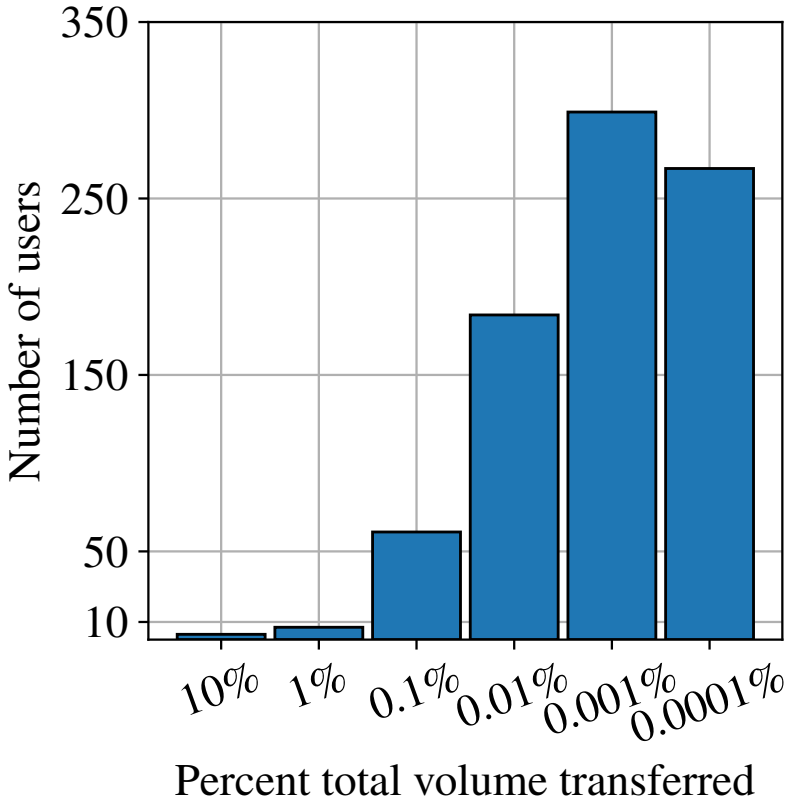
U.S. DEPARTMENT OF ENERGY | Office of Science

Argonne NATIONAL LABORATORY

BERKELEY LAB

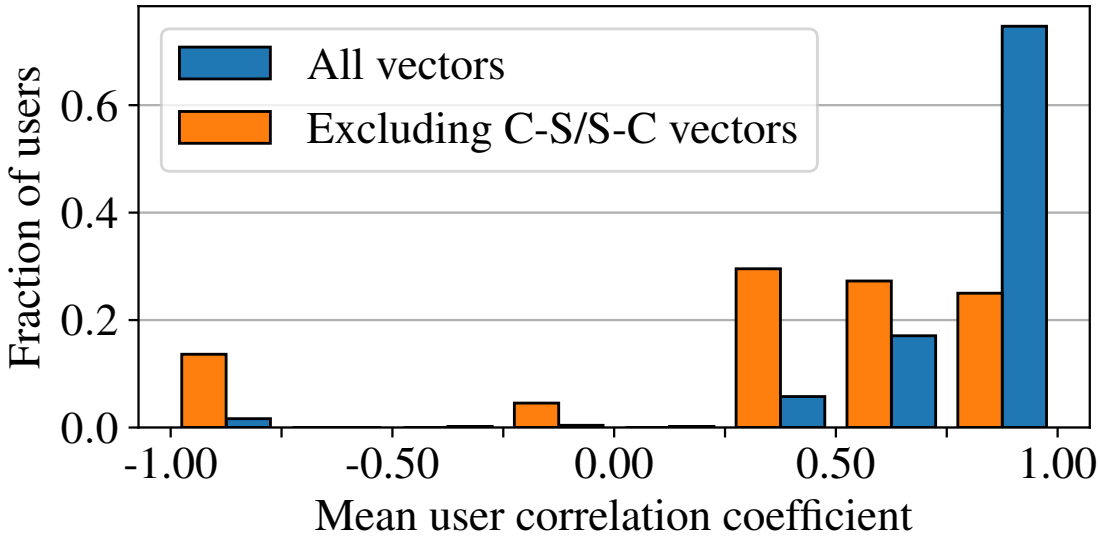
Few users result in the most transfers





Regularity of user I/O coupling

- **MUTC**
  - how correlatable is a user's I/O across all vectors
  - how easily we can guess what a user's workflow is doing
- **Strongest correlation only between job reads and job writes**
- **"Excluding C-S/S-C" only shows workflows with storage-storage or storage-WAN activity**



1,123 users represented in "all vectors"
486 users represented in "excluding C-S/S-C vectors"