Brookhaven National Laboratory

U.S. DEPARTMENT of ENERGY

Error Analysis of Globally Distributed Workflow Management Systems

Sankha Dutta, Computational Scientist
On behalf of the REDWOOD Collaboration

SC25 | St. Louis, MO

@BrookhavenLab

Presentation Roadmap

- Introduction
  - Workflow Management System (WMS)
  - A simple example
  - Different components of a WMS
- Motivation: WMS operational Complexity
  - Site
  - Data movement
  - Execution Core hours
  - Workload volume
- Understanding faults through analysis
- Conclusion

What is a Workflow Management System (WMS)?

- High-level interface for defining scientific workflows and tasks.

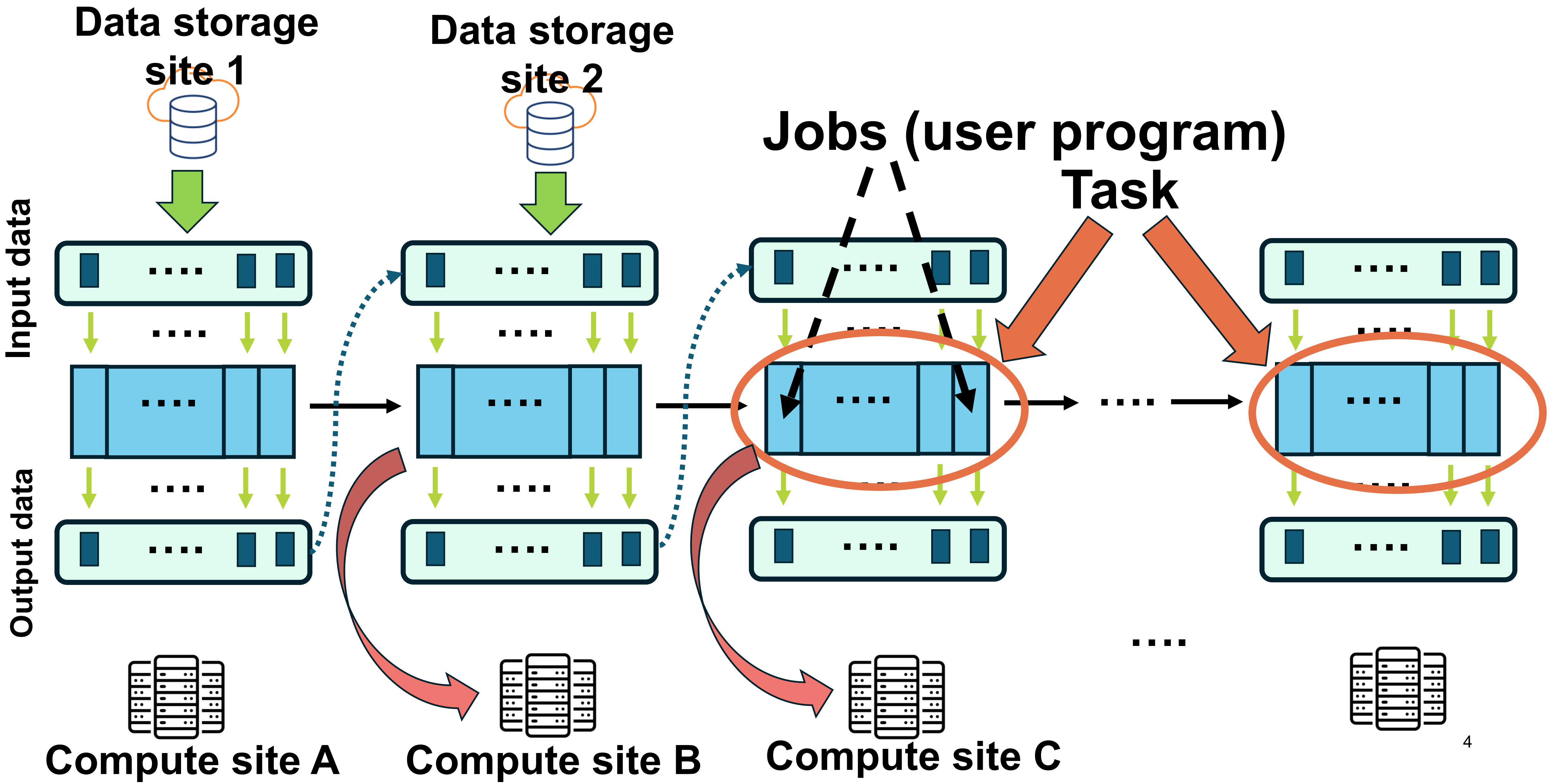
- One workflow corresponds to one scientific object submitted by user to computing sites.

- Automatically breaks tasks into jobs and dispatches to suitable sites.

- Handles data staging, execution coordination, and error reporting.

- Faults are common while managing such complex globally distributed computing systems.

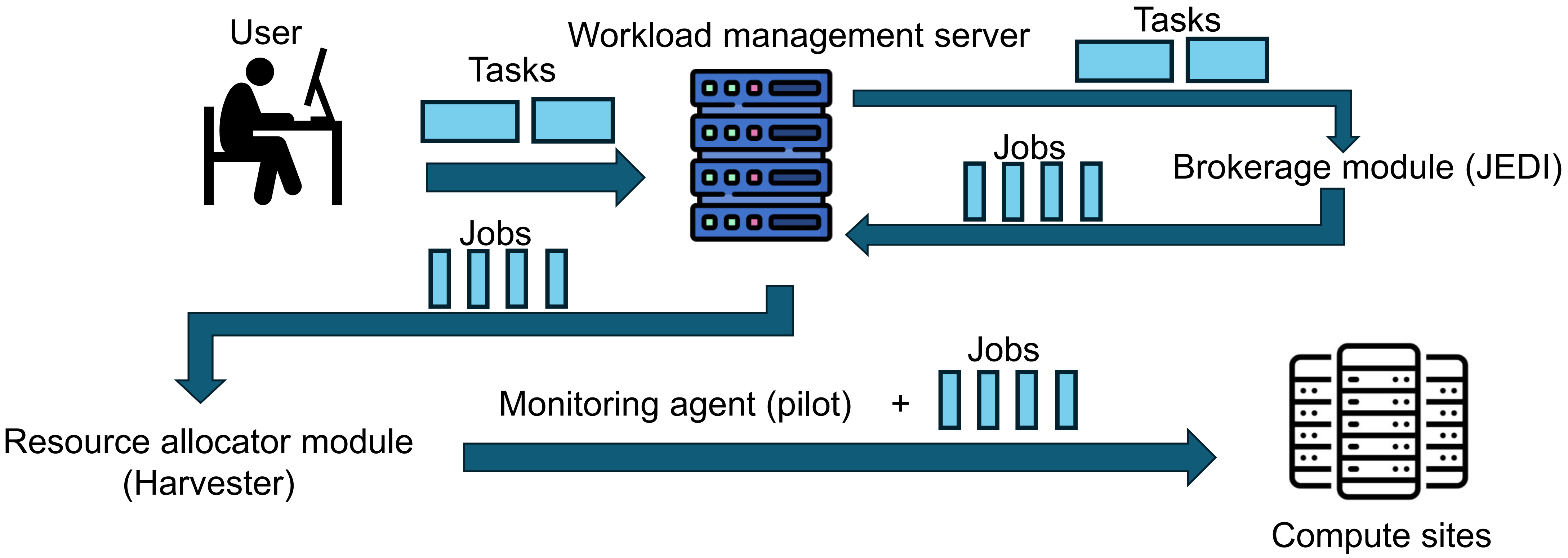
WMS Example


Data storage site 1
Data storage site 2
Jobs (user program)
Task
Input data
Output data
....
Compute site A
Compute site B
Compute site C

Managing Such Complex Frameworks

- WMS are highly modular with various components.
- Each is responsible for different functions.
- Our work is based on PanDA as a typical WMS.

Broad overview of different components in WMS



Managing Globally Distributed Systems is Challenging

Compute Site complexity

- Resources (nodes, cores, memories, storage, network, etc.) varies among compute sites.
- Different kinds of computing infrastructure, such as cloud, high-performance computing facilities and supercomputers.

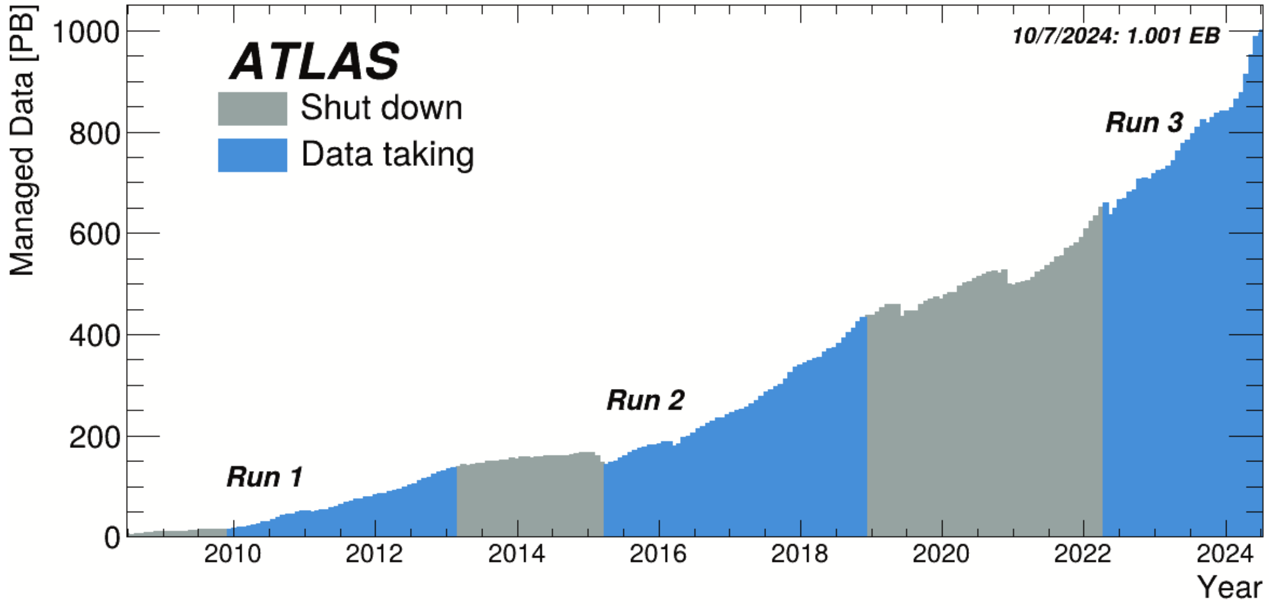
Number of Users

- User submits millions of workloads, which are then broken down for execution across different compute systems.
- Our data sample contains more than 50 million user programs.
- User programs from a single experiment (Vera C. Rubin) reached up to 60 million in July 2023.

Data Transfer in Global Systems

Steady increase of data transfer

- Data and their replicas are distributed globally across multiple sites.
- Global data transfer is reaching exabyte scale.



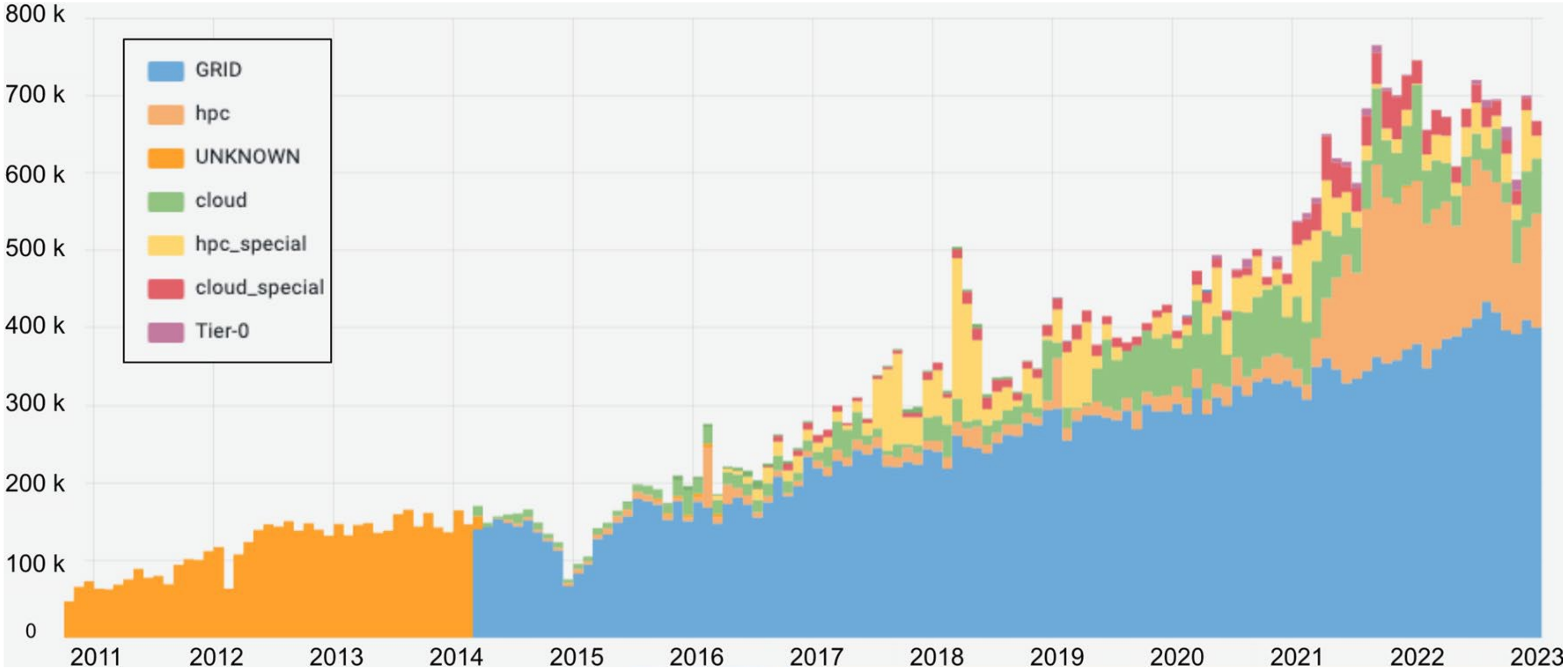
1 EB data transfer in 2024

ATLAS grid data transfer volume over years

Global Compute Resources Demand

High demand of compute resources

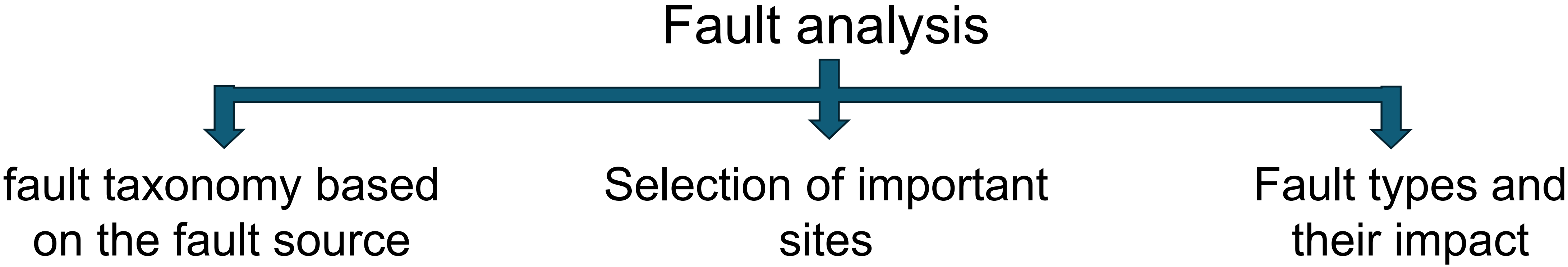
- Many CPUs are involved in job execution.
- Total daily CPU core hours are reaching up to 800K hours.



Monthly average CPU cores managed by PanDA WMS for ATLAS by resource types between 2011 and 2023.

Understanding Faults Through Analysis

- We encountered 12% faults in our data sample of 50 million user programs.
- Faults leading to suboptimal usage of core hours, unnecessary data movements, execution delay, etc.
- Important to understand faults → First step is fault analysis.



Fault analysis

fault taxonomy based on the fault source

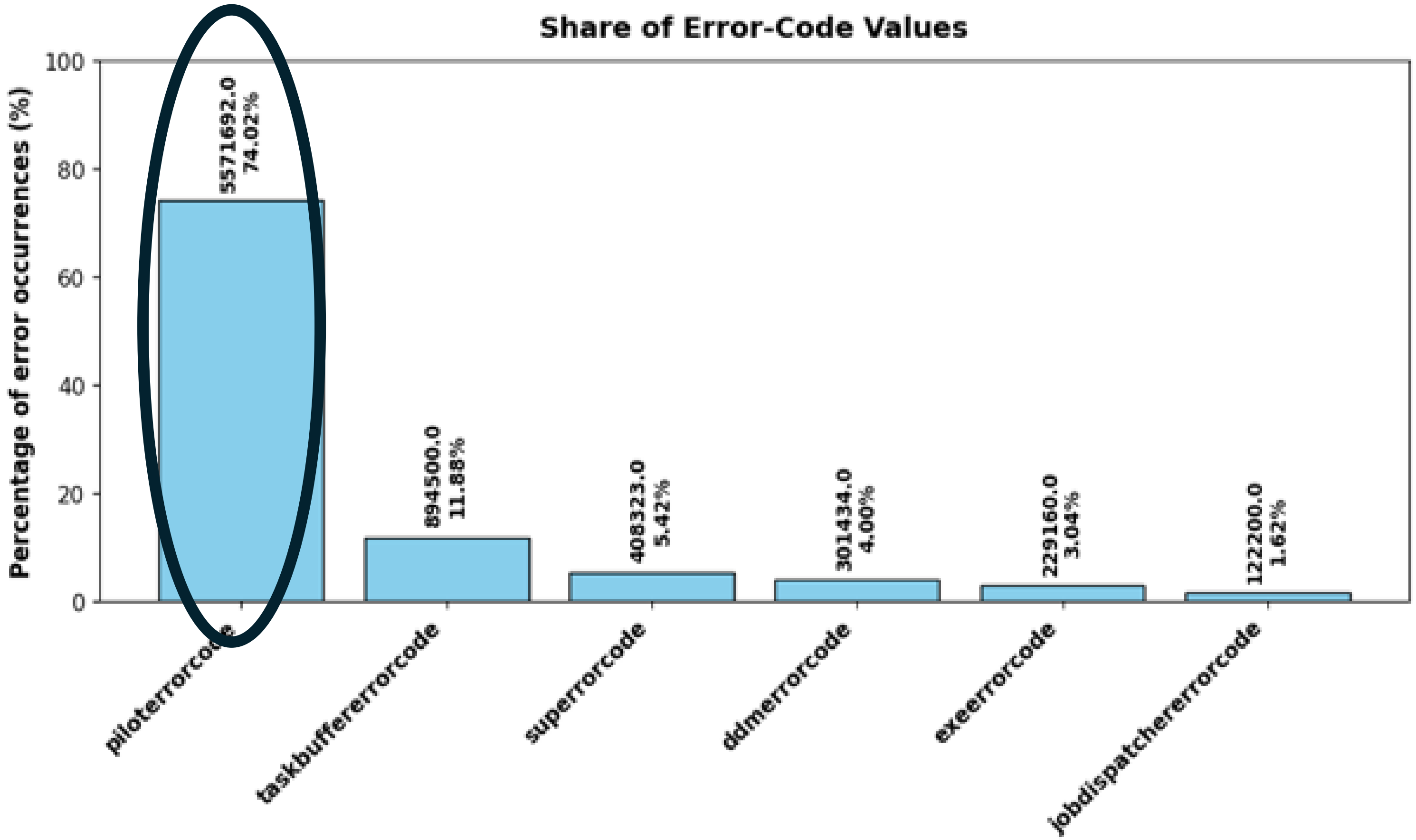
Selection of important sites

Fault types and their impact

Source-based Fault Taxonomy

Source	Description
Site monitoring module	Faults reported by the job health monitoring module on the compute site to the server.
Workload manager	Faults reported by the workload manager due to faults at the workload queue.
Data movement	Faults reported due to file transfers to execute jobs on the compute nodes.
Execution	Faults that arises from the scientific application itself submitted by the users to the compute nodes.
User program allocator	Faults occurring while assigning user programs to computing sites.
Resource allocation	Faults reported by the component responsible for resource allocation known as harvester.

Fault Distribution by Source



Pilot monitoring agent reports highest fault ≅ 74%

We focused on the faults reported by pilot agent only.

How are Sites Selected?

Identifying important attributes: Selected four attributes:

1. Number of user programs submitted (J_s)
2. Faults reported by site monitoring module (P_s)
3. User program queue wait time for each site (W_s)
4. User program execution time for each site (E_s)

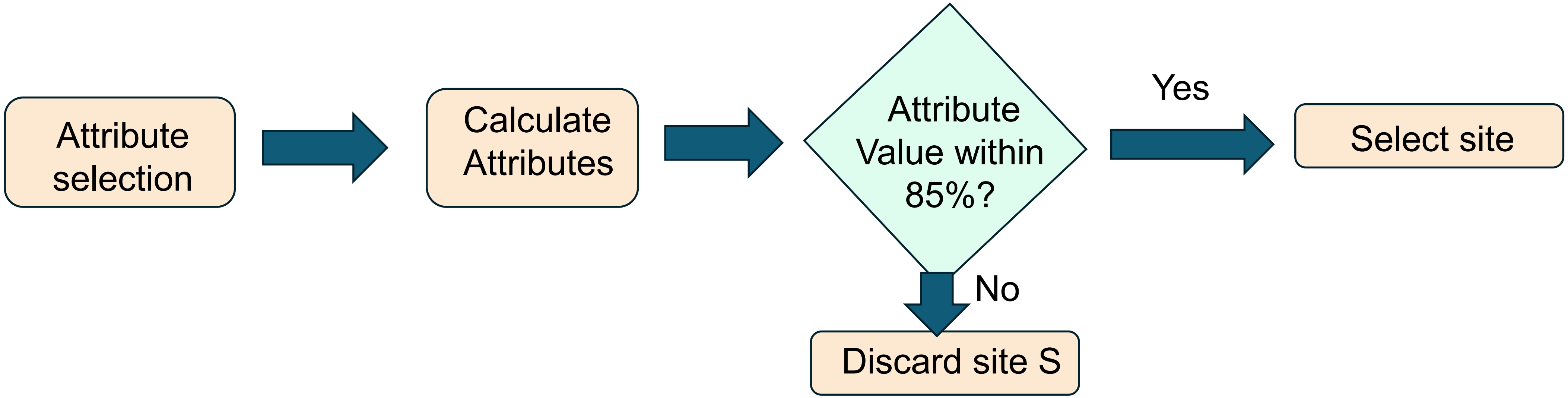
Calculating the share of the attributes per site:

$$\text{Attribute share per site } (J_s, P_s, W_s, E_s) = \frac{\text{sum of the attribute for a site S } (V_s)}{\text{Sum of the attribute over all sites } (C_s)}$$

e.g.,

$$J_s \text{ (share of the attribute number of user programs submitted for site S)} = \frac{\text{Sum of the number of user programs submitted to site S}}{\text{Sum of the number of user programs submitted to all sites}}$$

Site Selection Algorithm

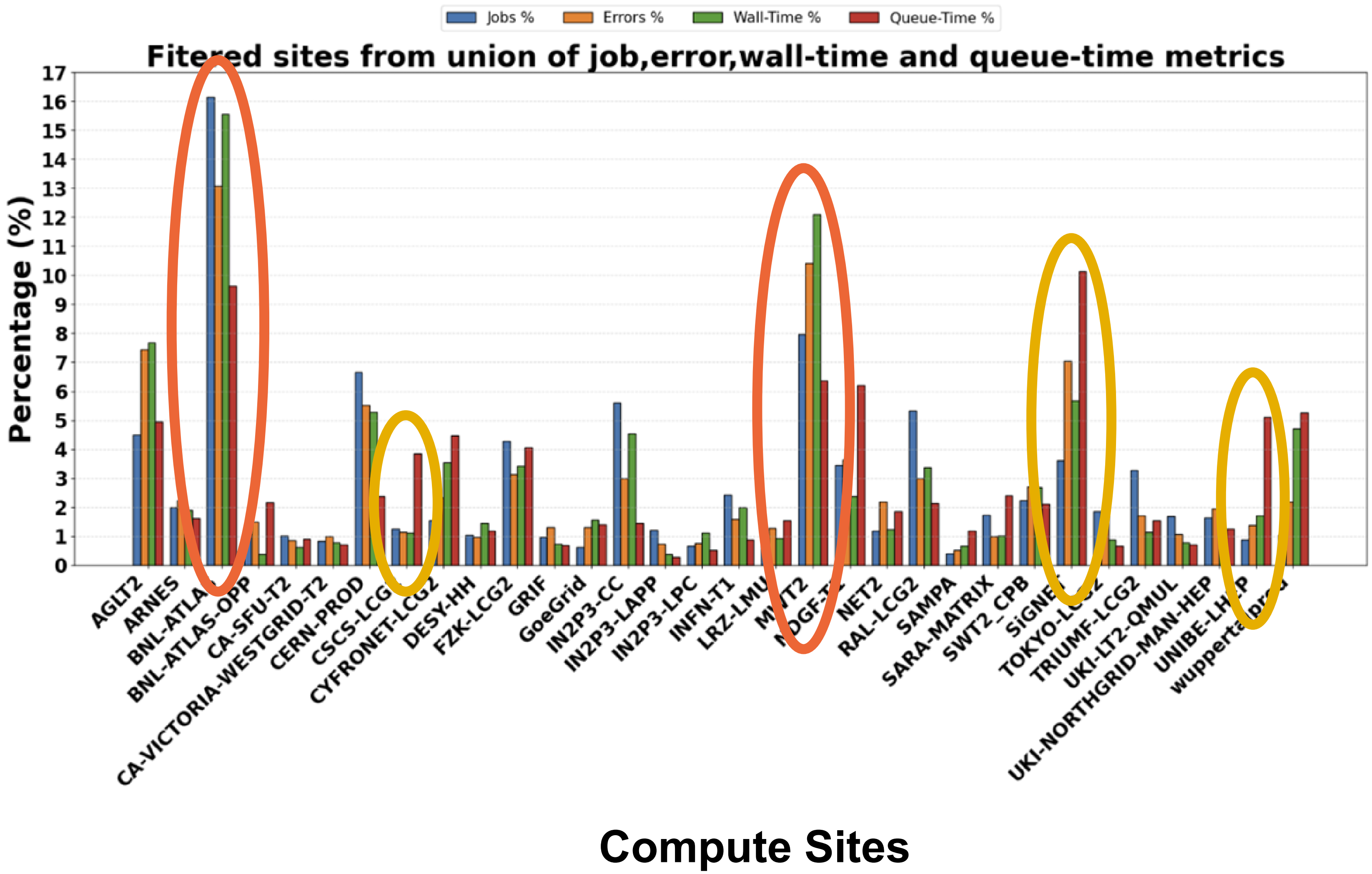


- Using each attribute, we get a selected site list for that attribute.
- Final sites are the union of selected sites from each attribute.

Selected sites = Job submission (J_s) ∪ Pilot error (P_s) ∪ Job wait time (W_s) ∪ Job execution time (E_s)

- We analyzed the final selected sites obtained through the union operation of each attribute.

Selected Sites based on Different Attributes



- 32 sites filtered sites from the 64 active sites from our dataset.
- Near 90% coverage of all parameters.
- Major sites such as site BNL_ATLAS or site MWT2 have high percentages in all four metrics.
- Some sites, e.g., site SiGNET, UNIBE_LHEP have much higher queue time compared to job execution time.

Fault Analysis by Types

- We studied faults reported by site monitoring module.
- Majority of faults reported by site monitoring module (74%).

Fault categories	Fault occurrence rationale
File	Local or remote file transfer or access faults.
Program termination	Program execution stopped because of explicit termination or stuck in a loop.
Resource limitation	Programs running out of resources or resources unavailable
Timeout failure	Programs exceeding allotted time during execution, transfer, etc.
Network issues	Unable to open remote file or service temporarily unavailable.
Execution failure	failed to launch or monitor the program.

- Execution failure covers the majority failure, often overlapping with other failures.

Impact of Faults on Resource Usage

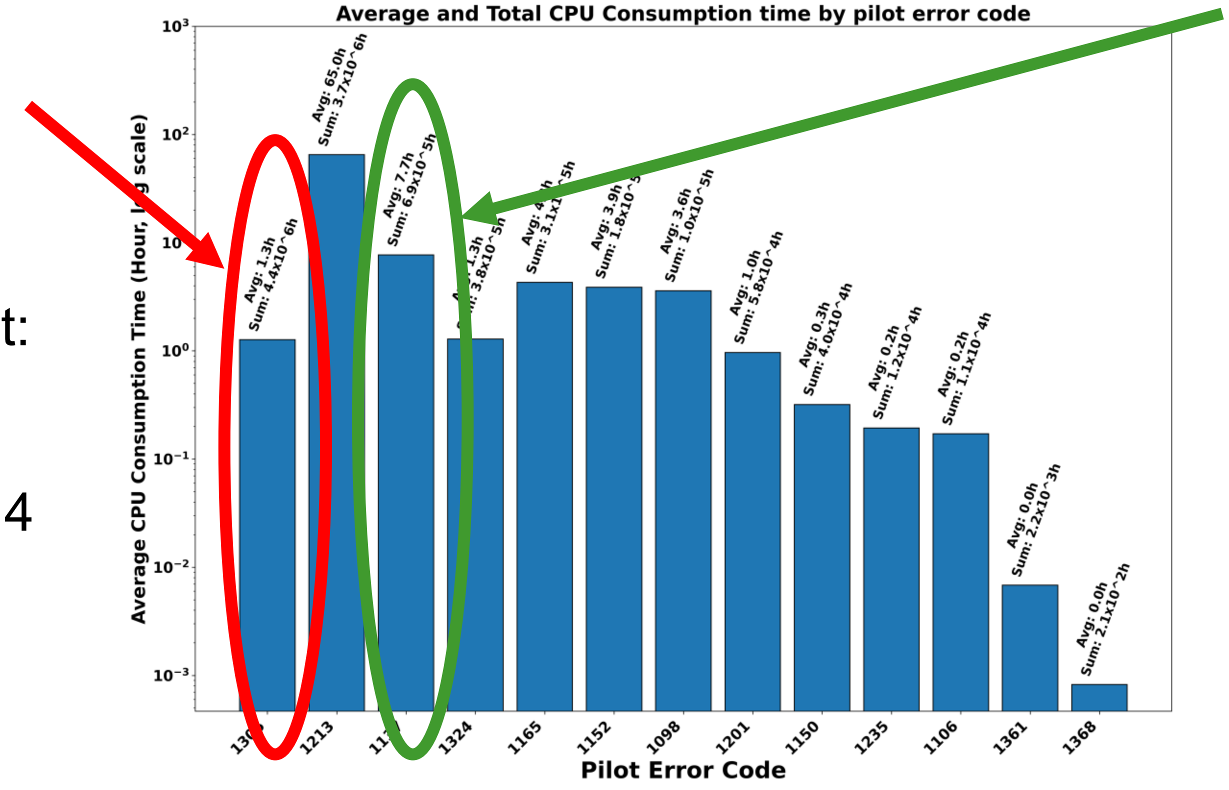
- Suboptimal use of resources due to faults. Some are more impactful than others.
- Used CPU consumption time as suboptimal resource usage.

Execution faults

Occurrence rate of 61%

Average core hours lost: 1.3 core Hours

Total core hours lost: 4.4 x10^6 core Hours



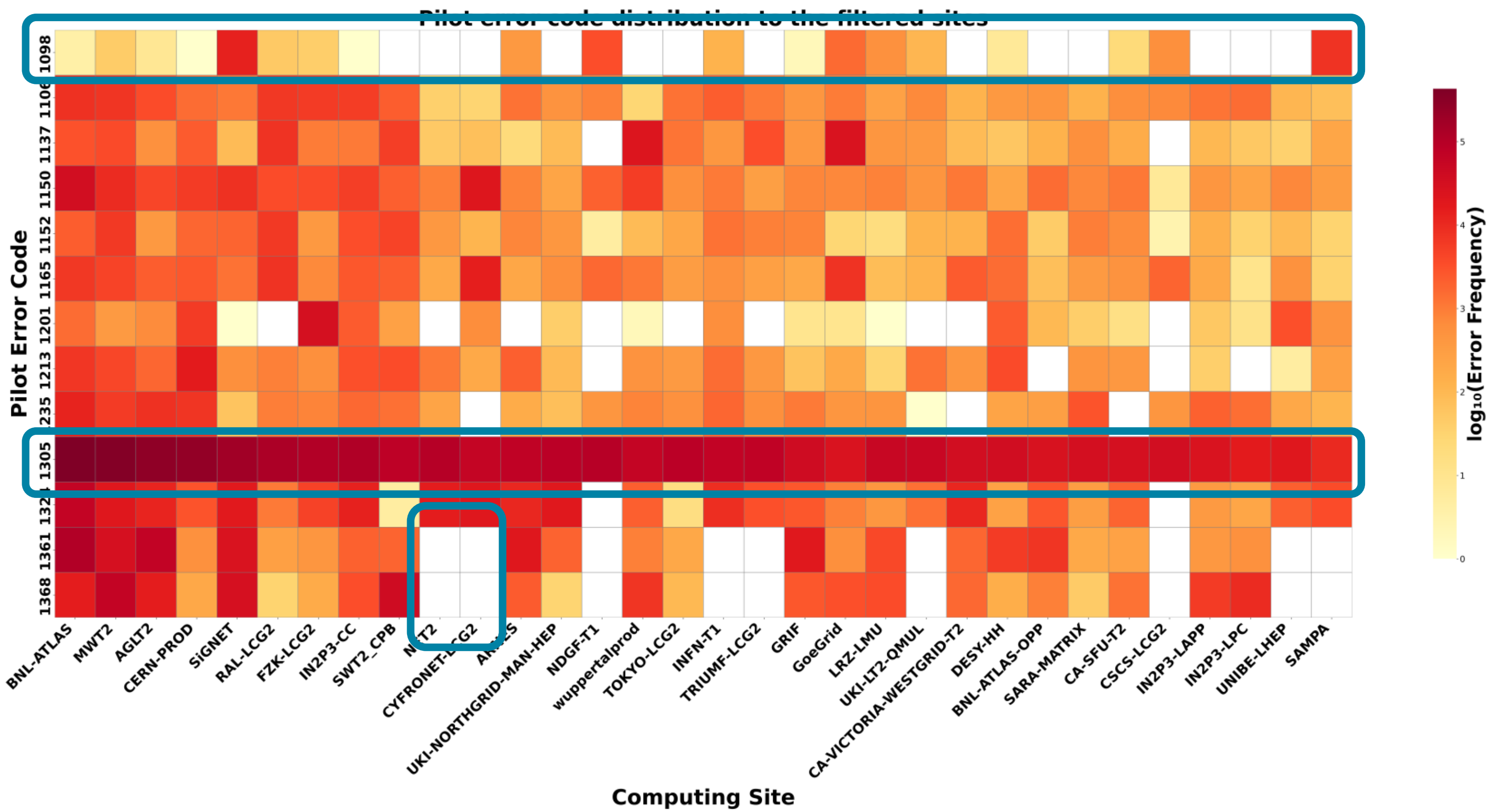
File issue

Occurrence rate of 1.7%

Average core hours lost: 7.7 core Hours

Total core hours lost: 6.9 x10^5 core Hours

Fault Distribution Across Sites



- Some faults are constant over all sites, e.g., user execution faults.
- Distribution shows relations between faults.
- Certain networks (unable to open remote file) leading to timeout faults are non-existant in some sites.
- Certain faults are low over all sites. E.g. Resource limit fault (1098)

Affords a holistic view of different faults over different sites for optimization and mitigation purposes.

Conclusion

- Globally distributed workflow system is complex with numerous components.

- Performed initial study of faults in complex globally distributed workflow systems.

- Our work:
  - Can categorize faults based on source
  - Designed methodology for fault analysis through site and fault selection
  - Impact of different faults on resources

- Our fault analysis is part of a larger ongoing project.

Thank you

Sankha Dutta, Ozgur O. Kilic, Tatiana Korchuganova, Paul Nilsson, Sairam Sri Vatsavai, Kuan-Chieh Hsu, David K. Park, Joseph Boudreau, Tasnuva Chowdhury, Shengyu Feng, Raees Khan, Jaehyung Kim, Scott Klasky, Tadashi Maeno, Verena Ingrid Martinez Outschoorn, Norbert Podhorszki, Yihui Ren, Frédéric Suter, Wei Yang, Yiming Yang, Shinjae Yoo, Alexei Klimentov, Adolfy Hoisie.

REDWOOD collaboration:

Brookhaven National Laboratory

Oak Ridge National Laboratory

University of Massachusetts Amherst 1863

Carnegie Mellon University

University of Pittsburgh Veritas Virtus 1787

SLAC National Accelerator Laboratory