CI/CS WORKSHOP

THE COMMUNITY TOGETHER

ResearchSOC | CI CoE PILOT

Mats Rynge: Workflows

USC Information Sciences Institute
www.isi.edu

Questions

- How pegasus takes care of protected data
- Have a few questions about feasibility of supporting workflow systems on a floating vessel

Trustworthy Data Working Group

Aims to provide guidance on data security for open science, to improve scientific productivity and trust in scientific results. Open science relies on data integrity, collaboration, high performance computing, and scalable tools to achieve results, but currently lacks effective cybersecurity programs that address the trustworthiness of scientific data.

Community Survey: Scientific Data Security Concerns and Practices

- 111 participants
- Report available: https://doi.org/10.5281/zenodo.3906865

PEARC'20 Workshop on Trustworthy Scientific Cyberinfrastructure

Next: creating a "Guidance for Science Projects and Cyberinfrastructure Developers" document

Protected Data?

Trustworthy Data?

Integrity - The data has not been altered.
Reproducibility - The data can be re-created, or the associated scientific results are replicable.
Provenance - The data's origin and lineage can be readily established.
Methodology - The processes and inputs used to create the data are well-established and accepted by the community.
Responsible stewardship - The ownership of the data is well managed and can be transferred.
Accuracy - The data is free from error.
Reputation - The data was generated by a credible or trusted source.
Significance - The data enables future research directions (with associated funding/support).
Availability - The data is there when I need it
Authorization - Way to vet and grant access
Confidentiality - Ensure repository hides/masks PII or other sensitive information from those not granted access
Accountability - Provision for metadata to describe the data, including provenance, versioning

SWIP
Scientific Workflow Integrity with Pegasus

The SWIP Project is supported by the National Science Foundation under Grant 1642070, 1642053, and 1642090.

Integrity Protection for Scientific Workflow Data: Motivation and Initial Experiences

Mats Rynge, Karan Vahi, Ewa Deelman **Information Sciences Institute - University of Southern California**

Anirban Mandal, Ilya Baldin **RENCI - University of North Carolina, Chapel Hill**

Omkar Bhide, Randy Heiland, Von Welch, Raquel Hill **Indiana University**

William L. Poehlman, F. Alex Feltus **Clemson University**

CENTER FOR APPLIED CYBERSECURITY RESEARCH

Pegasus

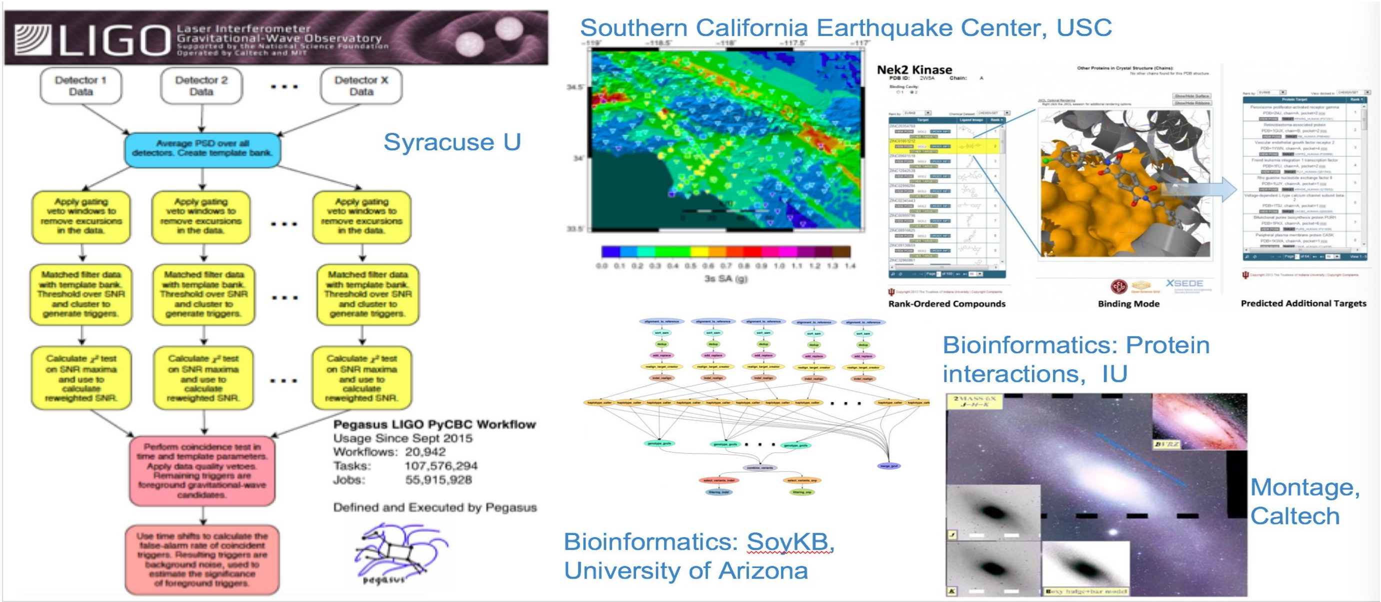
SICE

renci

Pegasus Workflow Management System, Production Use

Last 12 months: Pegasus users ran 240K workflows, 145M jobs

Majority of these include data transfers, using LAN, the Internet, local and remote storage



https://pegasus.isi.edu/

Scientific Workflow Integrity with Pegasus
NSF CICI Awards 1642070, 1642053, and 1642090

Goals:

Provide additional assurances that a scientific workflow is not accidentally or maliciously tampered with during its execution.

Allow for detection of modification to its data or executables at later dates to facilitate reproducibility.

Integrate cryptographic support for data integrity into the Pegasus Workflow Management System.

PIs: Von Welch, Ilya Baldin, Ewa Deelman, Raquel Hill
Team: Omkar Bhide, Rafael Ferrieira da Silva, Randy Heiland, Anirban Mandal, Rajiv Mayani, Mats Rynge, Karan Vahi

Our Talk

- **Introduction and Motivations**
- Our Approach
- Current Status
- Welcome to the Jungle
- Integrity Issues in the Wild
- Future Work

Data Integrity

0101110101010

0101000101010

Challenges to Scientific Data Integrity

Modern IT systems are not perfect - errors creep in.

At modern “Big Data” sizes we are starting to see checksums breaking down.

Plus there is the threat of intentional changes: malicious attackers, insider threats, etc.

User Perception: “Am I not already protected? I have heard about TCP checksums, encrypted transfers, checksum validation, RAID and erasure coding – is that not enough?”

Motivation: CERN/NEC Studies of Disk Errors

Examined Disk, Memory, RAID 5 errors.

“The error rates are at the 10^{-7} level, but with complicated patterns.” E.g. 80% of disk errors were 64k regions of corruption.

Explored many fixes and their often significant performance trade-offs.

A similar study by NEC found that 1 in 90 SATA drives will experience silent data corruption.

Data integrity

Bernd Panzer-Steindel, CERN/IT
Draft 1.3 8. April 2007

Executive Summary

We have established that low level data corruptions exist and that they have several origins. The error rates are at the 10^{-7} level, but with complicated patterns. To cope with the problem one has to implement a variety of measures on the IT part and also on the experiment side. Checksum mechanisms have to implemented and deployed everywhere. This will lead to additional operational work and the need for more hardware.

Introduction

During January and February 2007 we have done a systematic analysis of data corruption cases in the CERN computer center. The major work in the implementation of probes and automatic running schemes were done by Tim Bell, Olof barring and Peter Kelemen from the IT/FIO group. There have been similar problems reported in Fermilab and Desy and information exchange with them was done.
The following paper will provide results from this analysis, a judgment of the situation and a catalogue of measures needed to get the problem under control.
It is also to be seen as a starting point for further discussions with IT, the experiments and the T1 sites.

https://indico.cern.ch/event/13797/contributions/1362288/attachments/115080/163419/Data_integrity_v3.pdf
https://www.necam.com/docs/?id=54157ff5-5de8-4966-a99d-341cf2cb27d3

Motivation:
Network Corruption

Network router software inadvertently corrupts TCP data and/or checksum!

XSEDE and Internet2 example from 2013.

Second similar case in 2017: University of Chicago network upgrade caused data corruption for the FreeSurfer/Fsurf project.

Brocade TSB 2013-162-A

BROCADE

TECHNICAL SUPPORT BULLETIN

June 28, 2013

TSB 2013-162-A SEVERITY: Critical- Service Impact

PRODUCTS AFFECTED:
Brocade NetIron XMR/MLX 100G module (BR-MLX-100Gx2-X and BR-MLX-100Gx1-X).

CORRECTED IN RELEASE:
The fix will be in patch releases of NI 5.3.00eb, 5.4.00d and 5.5.00c and later releases.
This issue is not applicable to software release NI 5.2.00 and previous releases.

BULLETIN OVERVIEW

When transferring data through 100G modules, a portion of the packet may get corrupted.
Corruption is typically seen when transferring jumbo frames.

XSEDE
Extreme Science and Engineering Discovery Environment

HOME ABOUT USER SERVICES EDUCATIO

News

XSEDE Network Status

Posted by Bob Garza on 07/25/2013 18:27 UTC

On March 1, 2013 XSEDENet, the network between XSEDE Service Providers, moved to Internet2's Advanced Layer 2 Service (AL2S) national network to take advantage of new features and performance capabilities.

XSEDE was notified recently by Internet2 that an error was discovered on the devices that Internet2 uses on its AL2S network that could possibly lead to data corruption. This error could have affected approximately 0.001% of the data that traversed **each** AL2S device and was undetectable by the standard TCP packet checksum. These errors would have primarily affected data transfers using protocols that did not employ data integrity capabilities (application compression, encryption or checksums). XSEDE users who used secure copy (scp) to transfer files were not affected due to its application layer checksums. Data transfers initiated with the Globus Online web interface also were not affected as Globus Online implemented default checksums in December 2012. Other data transfers including manual gridftp or other protocols without data integrity checking could have been affected by this error.

By July 17, 2013 Internet2, in cooperation with the device vendor, upgraded all the affected devices with a new version of software that corrected the error. XSEDE recommends that users who transferred files using data transfer protocols that do not incorporate data integrity capabilities check the integrity of their file transfers that occurred between March 1, 2013 and July 17, 2013. Please refer to the XSEDE documentation on data integrity and validation of data transfers for details about data integrity checks.

Please submit any questions you may have by sending email to help@xsede.org or by submitting your questions through the XSEDE User Portal @ https://portal.xsede.org/help-desk.

https://www.xsede.org/news/-/news/item/6390

Motivation:
Software failures

Bug in StashCache data transfer software would occasionally cause silent failure (failed but returned zero).

Failures in the final staging out of data were not detected.

The workflow management system, believing workflow was complete, cleaned up. With the final data being incomplete and all intermediary data lost, ten CPU-years of computing came to naught.

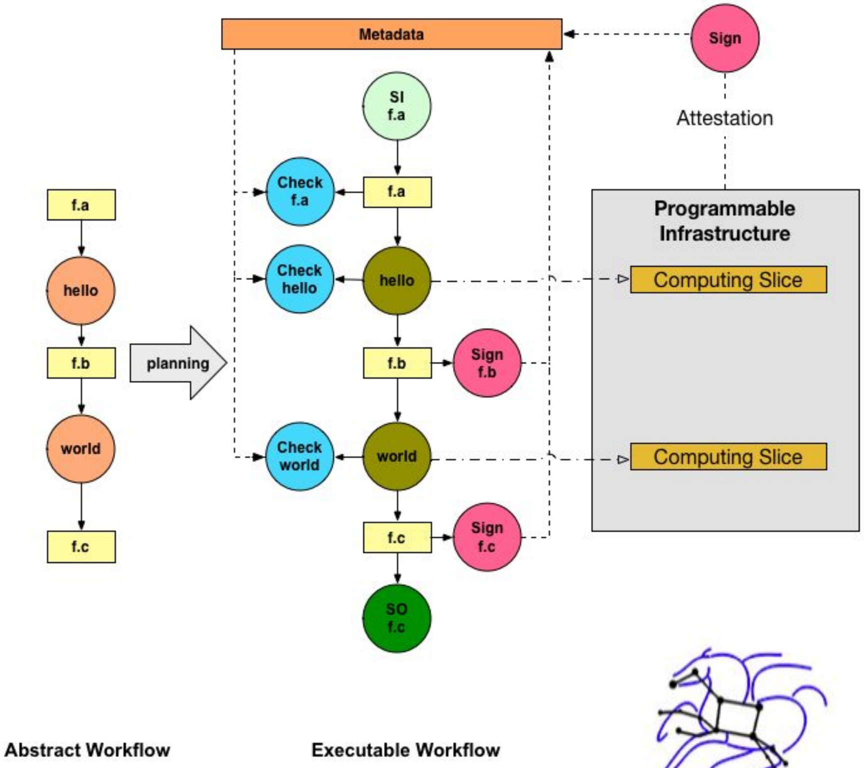
How is this an data integrity issue? The workflow system should have verified that the data at the storage system after the transfer, is the expected data.

Our Talk

- Introduction and Motivations
- **Our Approach**
- Current Status
- Welcome to the Jungle
- Integrity Issues in the Wild
- Future Work

Our High Level Plan...

- Workflow Management Systems (WMS) are great places to tackle data integrity.
- They understand what data is created and ingested and do not mind tedious tasks such as generating and checking checksums.
- Placement is important within the workflow of generate/validate checksums
- Pegasus WMS is widely used (LIGO, SCEC, SoyKB, Montage, etc.) by the scientific community and is the target of our improvements.



Application-level Checksums – SHA256

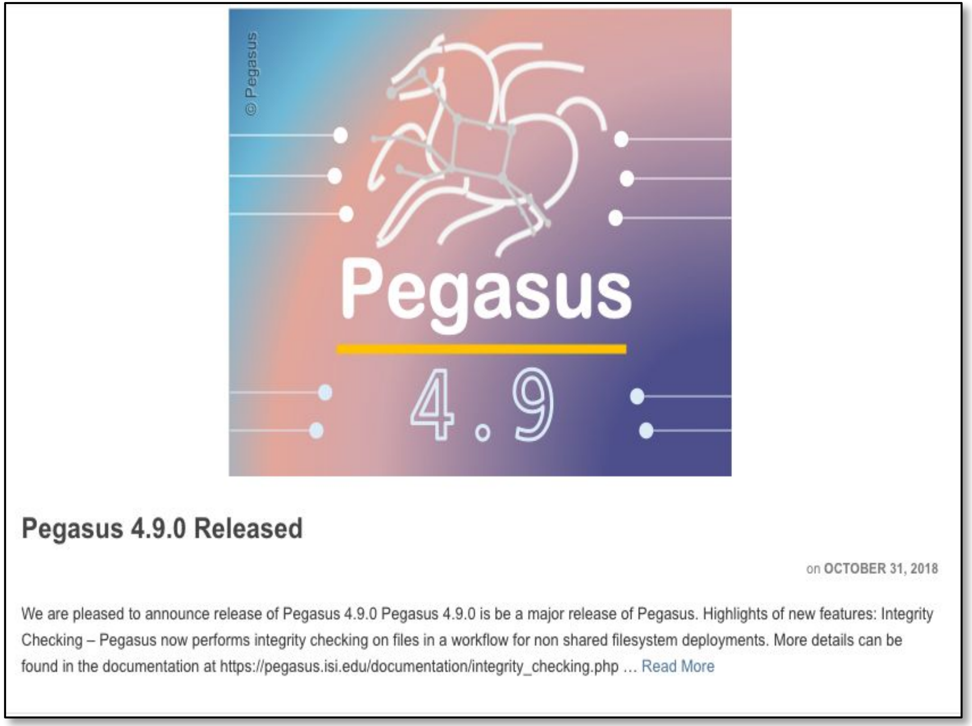
- Application-level checksums (hashes) allow for detection of changes.
- Explored some more advanced solutions, but at the end simplicity won
- Checksums already in use by many data transfer applications: scp, Globus/GridFTP, some parts of HTCondor, etc, but SWIP is focusing on end-to-end as well as over longer time periods

e.g. using a SHA in Python:

```
>>> hashlib.sha256(b"The Answer to the Ultimate Question of Life, the Universe, and Everything is 42").hexdigest()
'8a72856cf94464dd641f0a2620ab604dd7a3f50293784a3a399acf6dc5b651cb'

>>> hashlib.sha256(b"The Answer To the Ultimate Question of Life, the Universe, and Everything is 42").hexdigest()
'a39be9fd272f2569aa95a07134a55f032ecb5c51cef6d66fe4032ec30bf4f1b6'

>>> hashlib.sha256(b"The Answer is 42").hexdigest()
'cbf296e175f02156cd60d6bf93aebd92893e72a0c4c48eadef092d0dc7e28fc1'
```

Our Talk

- Introduction and Motivations
- Our Approach
- **Current Status**
- Welcome to the Jungle
- Integrity Issues in the Wild
- Future Work



Pegasus 4.9.0 Released

on OCTOBER 31, 2018

We are pleased to announce release of Pegasus 4.9.0 Pegasus 4.9.0 is be a major release of Pegasus. Highlights of new features: Integrity Checking – Pegasus now performs integrity checking on files in a workflow for non shared filesystem deployments. More details can be found in the documentation at https://pegasus.isi.edu/documentation/integrity_checking.php … Read More

Integrity validation is on by default since the Pegasus 4.9.0 release (Oct 31st, 2018). Users who upgrade will automatically get the protection, but can opt out.

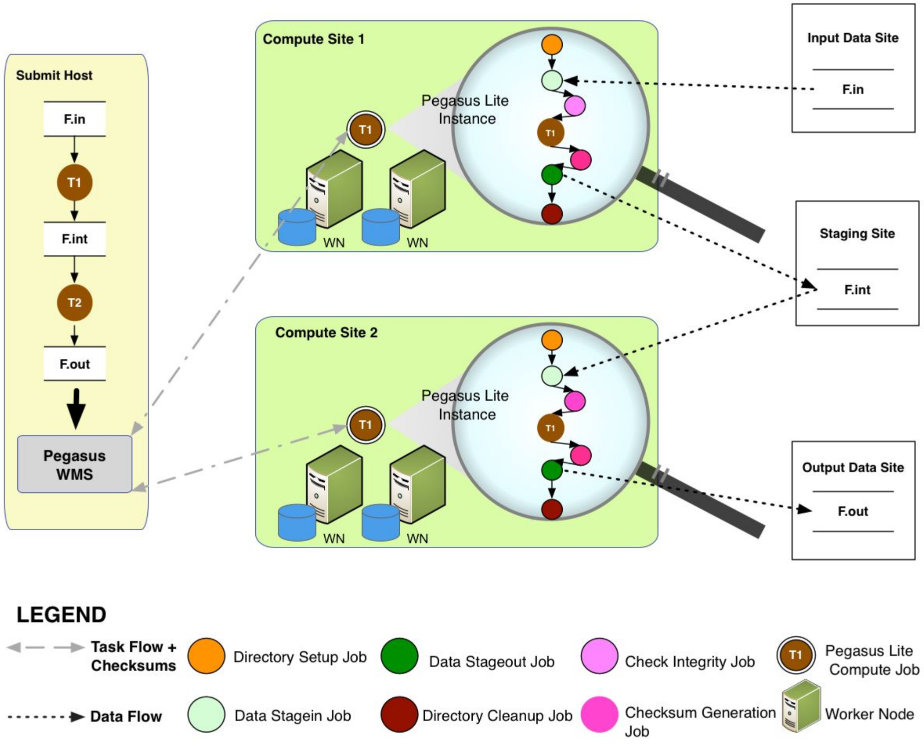
Sharing of detailed monitoring data with the Pegasus team is off by default. Users can opt-in. (We will come back to this at the end of the talk)

Automatic Integrity Checking in Pegasus

Pegasus performs integrity checksums on input files right before a job starts on the remote node.

- For raw inputs, checksums specified in the input replica catalog along with file locations
- All intermediate and output files checksums are generated and tracked within the system.
- Support for sha256 checksums

Job failure is triggered if checksums fail



Our Talk

- Introduction and Motivations
- Our Approach
- Current Status
- **Welcome to the Jungle**
- Integrity Issues in the Wild
- Future Work

How do you know your integrity protection is working?

- <u>Imagine the following:</u>
  You finish adding integrity protection to your software. You run a workflow and all goes smoothly.
- Was there no integrity problem or did you just fail to detect it?
- How do you reliably and repeatedly test integrity protection?

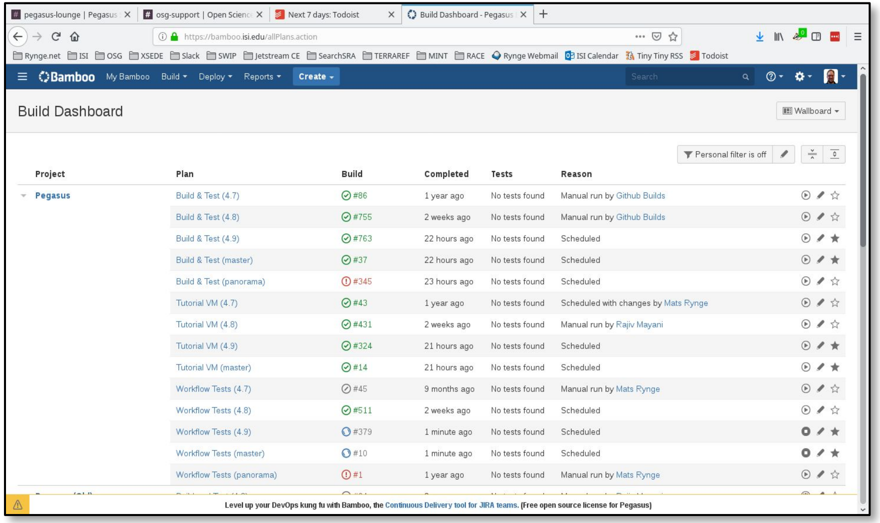
Confidence in the implementation: Bamboo

- **At commit, for each target platform:**

1. Build binary, workers, RPMs, DEBs, ...
2. Run unit tests for Java, Python, and C components
3. ~ 100 unit tests

- **Nightly:**

1. Run functional tests. These are full workflows, configured to provide good code coverage
2. ~ 85 workflows



Enter the Chaos Jungle!

https://github.com/RENCI-NRIG/chaos-jungle

Inspired by Netflix's Chaos Monkey.
https://github.com/Netflix/chaosmonkey

Goal of Chaos Jungle (CJ) is to introduce different kinds of impairments into the virtual infrastructure - network, compute, storage.

The RENCI ORCA software creates virtual infrastructure on ExoGENI testbed. CJ software introduces impairments into data transfers.

We get virtual infrastructure that intentionally corrupts data

Randomly or predictably?

Now we can test how software runs under bad conditions.

https://commons.wikimedia.org/wiki/File:Tioman_Rainforest.JPG

Chaos Jungle

Uses Linux eBPF (extended Berkeley Packet Filters) functionality

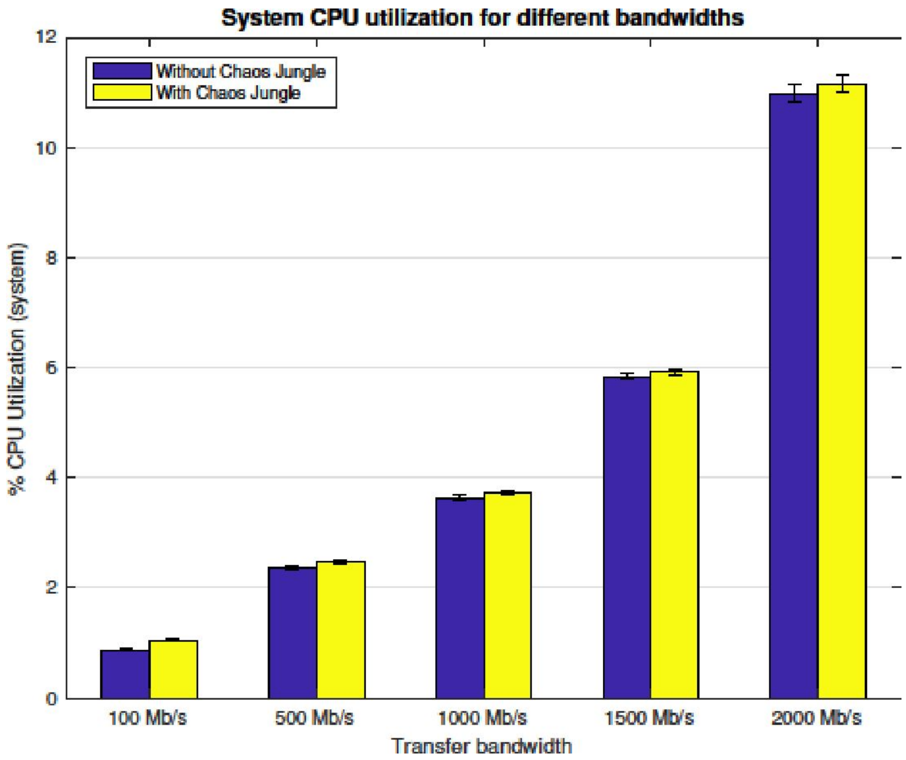
Introduces a small eBPF program into the kernel attaching to either TC filter or XDP hooks

Inspects received packets and modifies some of those that match flow descriptors without affecting the appropriate checksums.

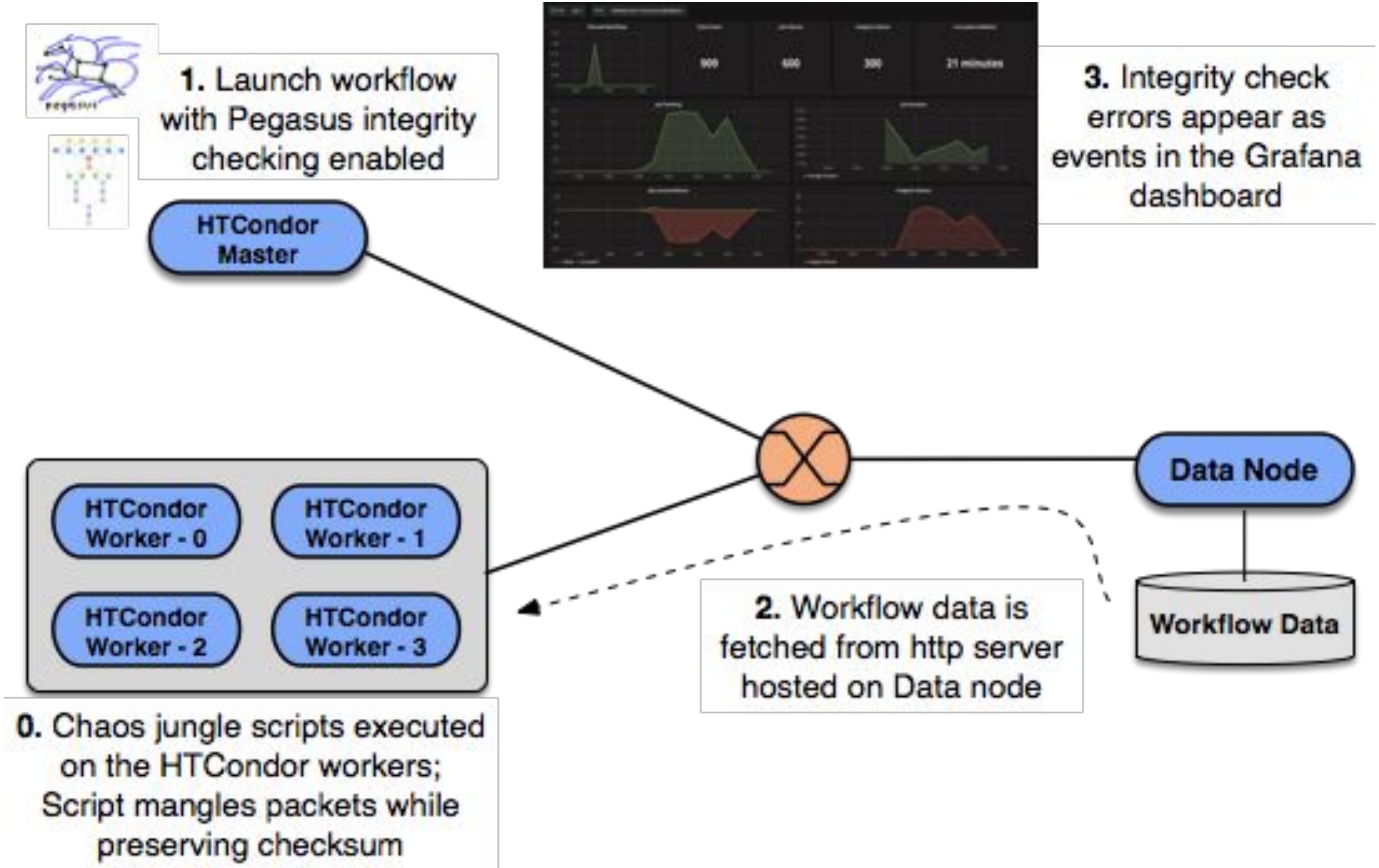
The packets thus look valid on the receiving end, however contain invalid data.

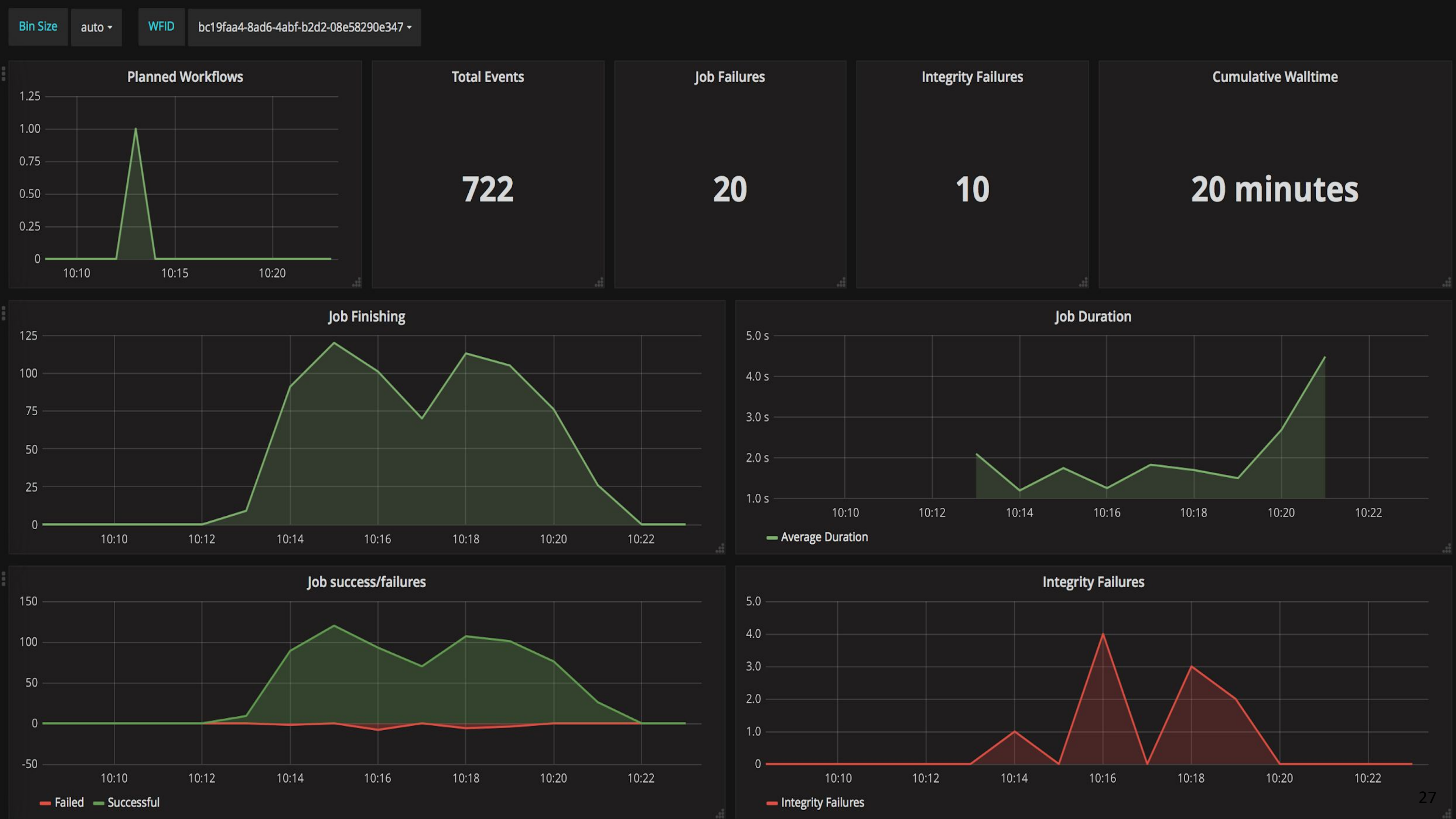
Fast and performant.

https://github.com/RENCI-NRIG/chaos-jungle



Chaos Jungle Experiment Setup



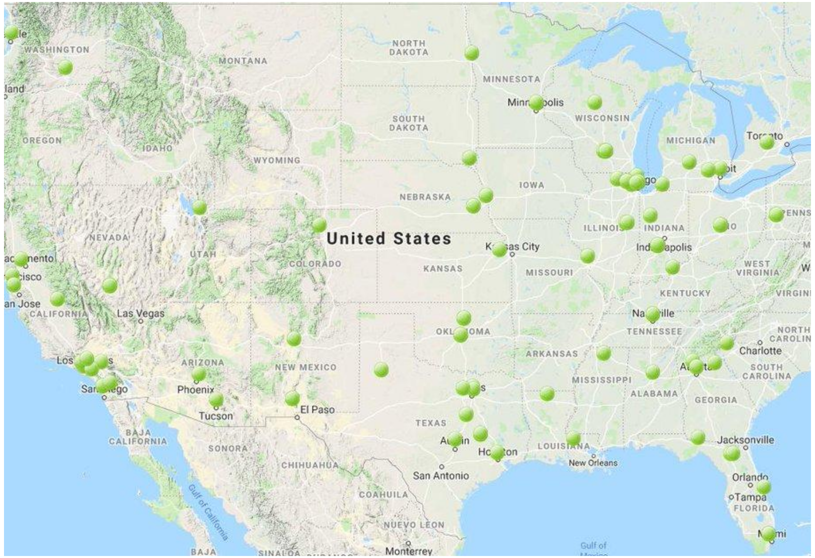

Bin Size
auto
WFID
bc19faa4-8ad6-4abf-b2d2-08e58290e347
Planned Workflows
1.25
1.00
0.75
0.50
0.25
0
10:10
10:15
10:20
Total Events
722
Job Failures
20
Integrity Failures
10
Cumulative Walltime
20 minutes
Job Finishing
125
100
75
50
25
0
10:10
10:12
10:14
10:16
10:18
10:20
10:22
Job Duration
5.0 s
4.0 s
3.0 s
2.0 s
1.0 s
10:10
10:12
10:14
10:16
10:18
10:20
10:22
Average Duration
Job success/failures
150
100
50
0
-50
10:10
10:12
10:14
10:16
10:18
10:20
10:22
Failed
Successful
Integrity Failures
5.0
4.0
3.0
2.0
1.0
0
10:10
10:12
10:14
10:16
10:18
10:20
10:22
Integrity Failures

Our Talk

- Introduction and Motivations
- Our Approach
- Current Status
- Welcome to the Jungle
- **Integrity Issues in the Wild**
- Future Work

Production Workflows

- Large workflows with lots of data transfers
- "Unprotected" protocols - no SSL or other protocol level protections
- Open Science Grid - WAN transfers
- Collecting the data is on an opt-in basis



Initial Results with Integrity Checking on

- OSG-KINC workflow (50,606 jobs) encountered 60 integrity errors in the wild (production OSG). The problematic jobs were automatically retried and the workflow finished successfully.
- The 60 errors took place on 3 different hosts. The first one at UColorado, and group 2 and 3 at UNL hosts.
- Error Analysis (by hand)
  - 1 input file error at University of Colorado.
  - 3 input file (kinc executable) errors on one node at University of Nebraska. The timespan across the failures was 16 seconds. We suspect that the node level cache got corrupted.
  - 56 input file errors on a different compute nodes at University of Nebraska. The timespan across the failures was 1,752 seconds. We suspect that the site level cache got corrupted.

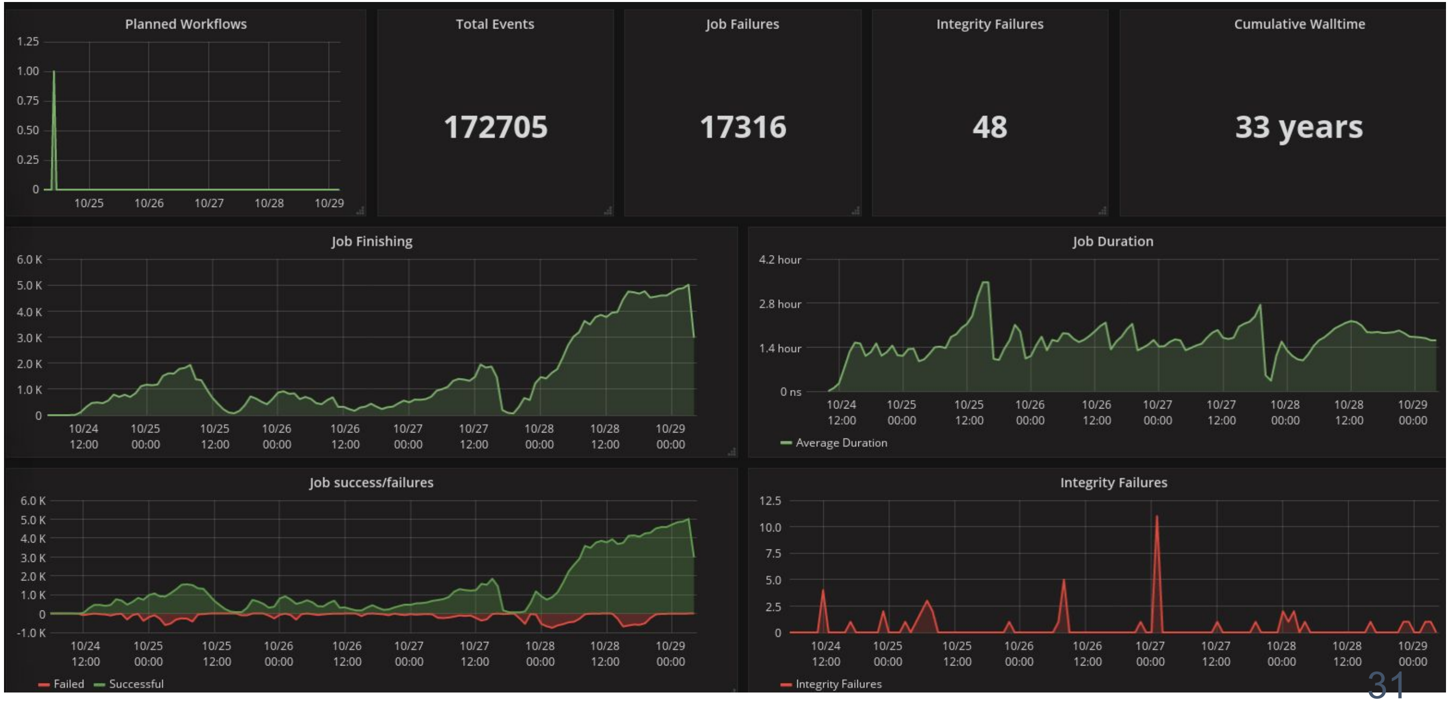
Initial Results – VERITAS / Nepomuk Otte, GATech

Seeing very small, but steady stream of corrected integrity errors from reporting back to Pegasus dashboard.

For VERITAS, ~.04% of transfers fail with integrity errors. (~1 / 2500 transfers)

Cause uncertain (diagnosis is harder than detection).

Possibly errors in http based transfers (s3 protocol against CEPH)



Checksum Overheads

- We have instrumented overheads and are available to end users via pegasus-statistics.

```
Type          Succeeded  Failed  Incomplete  Total   Retries  Total+Retries
Jobs          1606       0       0           1606    31       1637

Workflow wall time                                   : 7 hrs, 59 mins
Cumulative job wall time                             : 17 days, 23 hrs
# Integrity Metrics
3944 files checksums compared with total duration of 9 mins, 18 secs
1947 files checksums generated with total duration of 4 mins, 37 secs
# Integrity Errors
Failures: 0 jobs encountered integrity errors
```

1000 Node OSG Kinc Workflow
Overhead of 0.054 % incurred

- Other sample overheads on real world workflows
  - Ariella Gladstein's population modeling workflow
    - A 5,000 job workflow used up 167 days and 16 hours of core hours, while spending 2 hours and 42 minutes doing checksum verification, with an overhead of 0.068%.
  - A smaller example is the Dark Energy Survey Weak Lensing Pipeline with 131 jobs.
    - It used up 2 hours and 19 minutes of cumulative core hours, and 8 minutes and 43 seconds of checksum verification. The overhead was 0.062%.

Our Talk

- Introduction and Motivations
- Our Approach
- Current Status
- Welcome to the Jungle
- Integrity Issues in the Wild
- **Future Work**

Challenges

- Can we do more than know "something changed?"
- Detecting error easier than diagnosing error.
- Balance performance / integrity trade-off?
- How do we handle storage without compute capabilities?
- Long data life: today's cryptographic algorithms will probably not last as long as we need the science data.
  E.g. what threats will Quantum computing bring?
- When do we hit limits of cryptographic algorithms (collisions)?
- Are all errors in all types of data of equal concern?

Going Forward: Integrity Introspection for Scientific Workflows (IRIS)

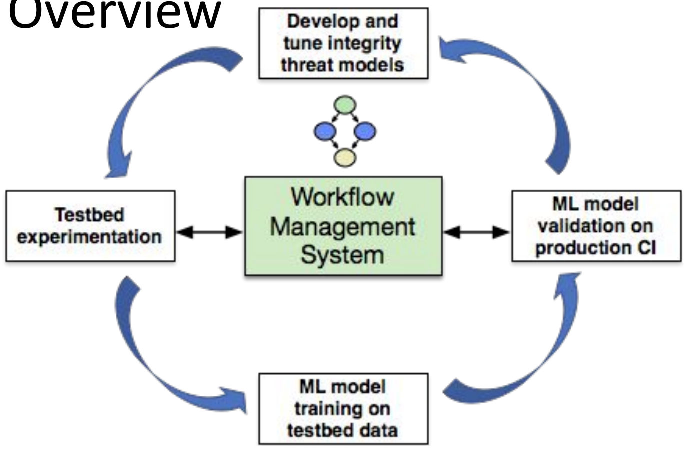
- National Science Foundation CICI IRIS Grant #1839900

- SWIP addresses ***integrity checking*** making sure that workflow computations are protected from integrity errors, but
  - Doesn't address analysis of integrity errors discovered, i.e. tracing the source of error or doing root cause analysis to remedy the problem.

- IRIS goal: Detect, diagnose, and pinpoint the source of unintentional integrity anomalies in scientific workflow executions on distributed cyberinfrastructure. ***(integrity analysis)***

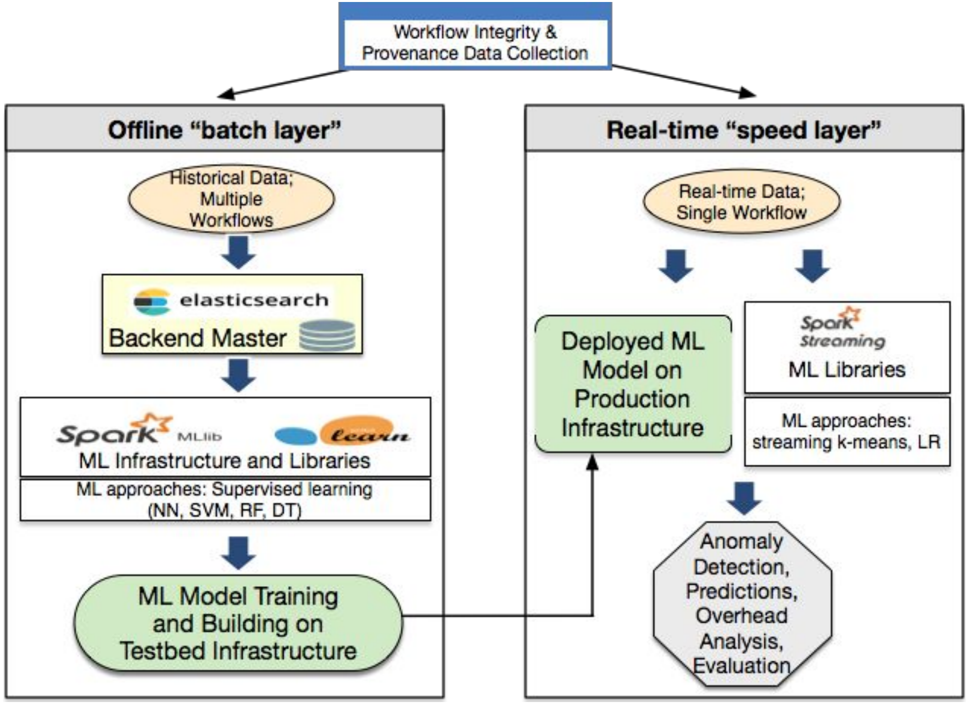
IRIS Overall Approach

IRIS Overview



Train ML algorithms on controlled testbeds and validate on national CI by integrating framework with Pegasus.

Engage with science application partners to deploy the analysis framework.



IRIS proposed framework

Thanks!

We thank the National Science Foundation for funding this work (Grants 1642070, 1642053, 1642090). Views expressed may not necessarily be the views of the NSF. Thanks to Eli Dart for Brocade TSB details.

Pegasus - a dHTC friendly workflow manager

Mats Rynge
rynge@isi.edu

USC Viterbi
School of Engineering
Information Sciences Institute

https://pegasus.isi.edu

Pegasus Concepts

Users describe their pipelines in a portable format called Abstract Workflow, without worrying about low level execution details.

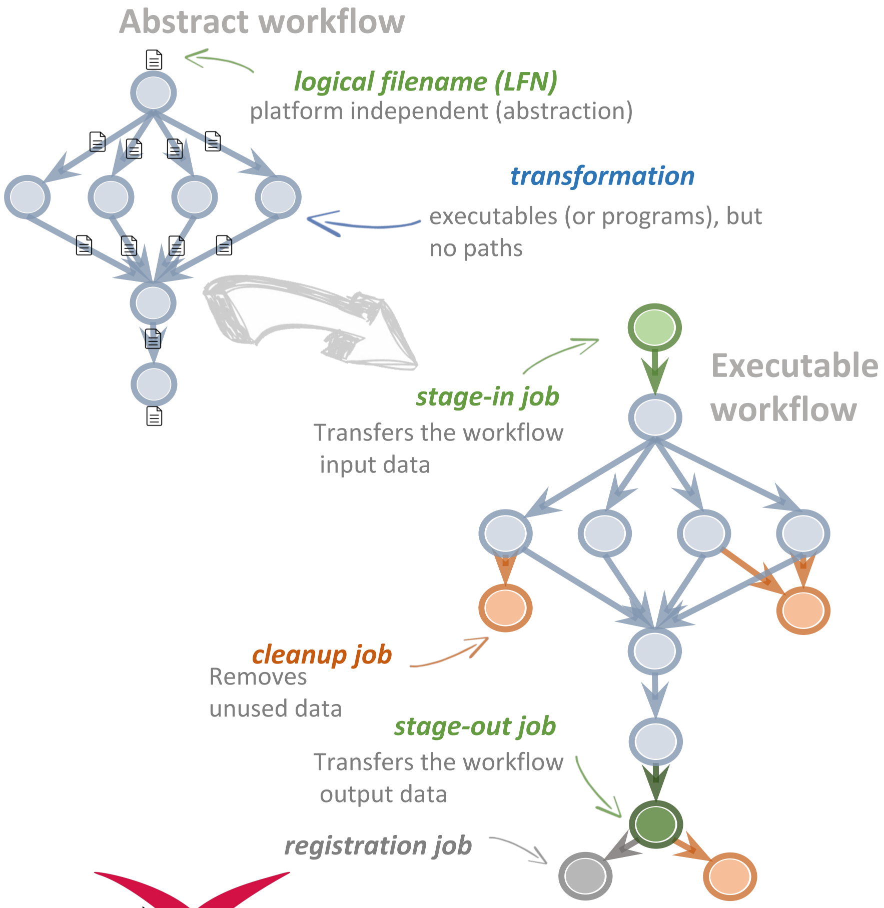
Workflows are DAGs

- Nodes: jobs, edges: dependencies
- No while loops, no conditional branches
- Jobs are standalone executables
- All data is tracked

Pegasus takes this and generates an executable workflow

- Data management tasks added
- Transforms the workflow for performance and reliability
- HTCondor DAGMan DAG

Planning occurs before execution



HTCondor
High Throughput Computing

Pegasus 5.0

Automate, recover, and debug scientific computations

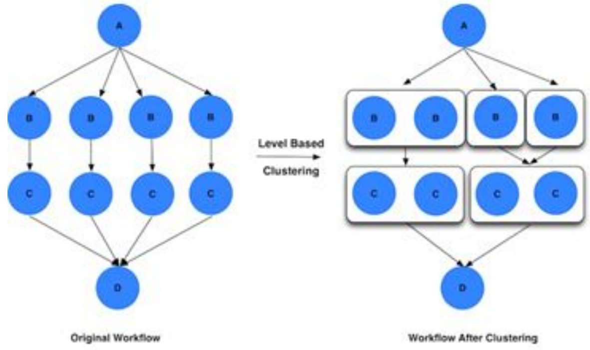
Coming soon! Beta1 is out.

- New and fresh Python3 API to compose, submit and monitor workflows, and configure catalogs
- New Catalog Formats
- Python 3
  - All Pegasus tools are Python 3 compliant
  - Python PIP packages for workflow composition and monitoring
- Zero configuration required to submit to local HTCondor pool.
- Data Management Improvements
  - New output replica catalog that registers outputs including file metadata such as size and checksums
  - Improved support for hierarchical workflows
- Major documentation improvements
  - https://pegasus.isi.edu/docs/5.0.0dev/index.html

```
#!/usr/bin/env python3
import logging
import sys

from Pegasus.api import *

# logs to be sent to stdout
logging.basicConfig(level=logging.DEBUG, stream=sys.stdout)

# --- Transformations ---------------------------------------
echo = Transformation(
        "echo",
        pfn="/bin/echo",
        site="condorpool"
    )

tc = TransformationCatalog()\
        .add_transformations(echo)

# --- Workflow -----------------------------------------------
Workflow("hello-world", infer_dependencies=True)\
    .add_jobs(
        Job(echo)
            .add_args("Hello World")
            .set_stdout("hello.out")
    ).add_transformation_catalog(tc)\
    .plan(submit=True)\
    .wait()
```

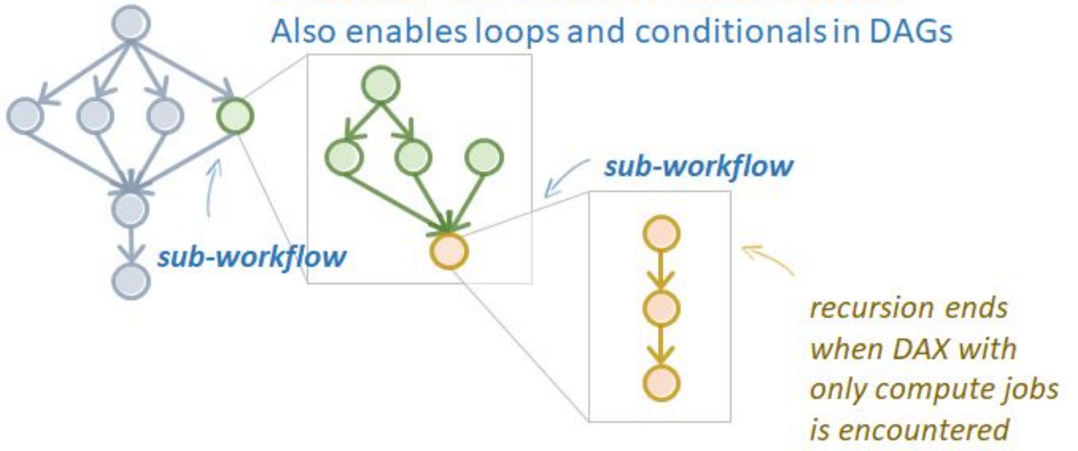
Optimizations

Task clustering

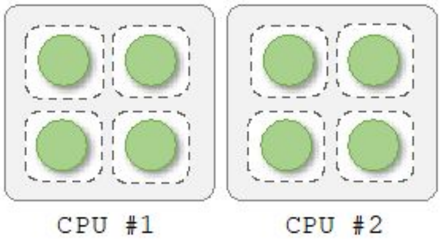


Hierarchical workflows

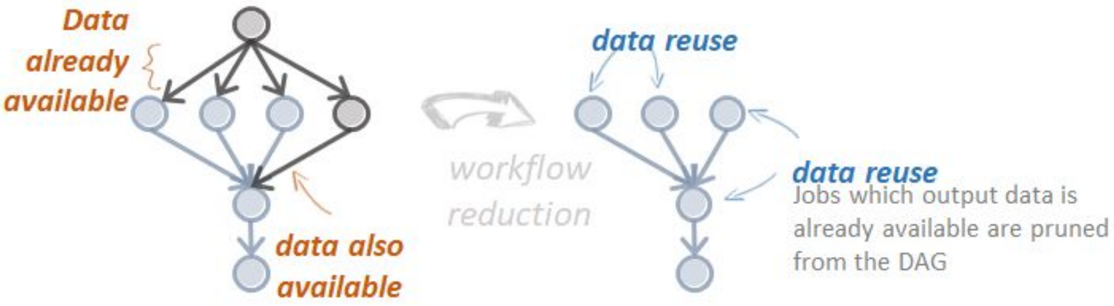
Enacts the execution of **millions of tasks**
Also enables loops and conditionals in DAGs



Task-resource co-allocation



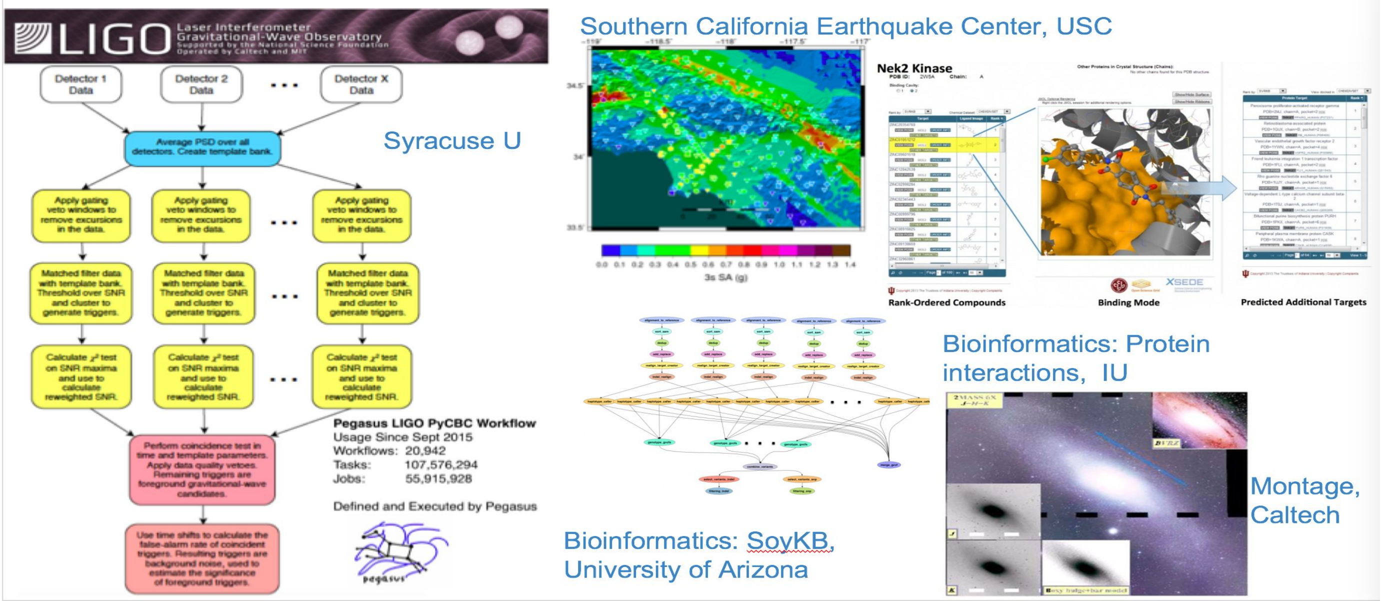
Data Reuse



Pegasus Workflow Management System, Production Use

Last 12 months: Pegasus users ran 240K workflows, 145M jobs

Majority of these include data transfers, using LAN, the Internet, local and remote storage



https://pegasus.isi.edu/

Data Staging Configurations

HTCondor I/O (HTCondor pools, OSG, …)

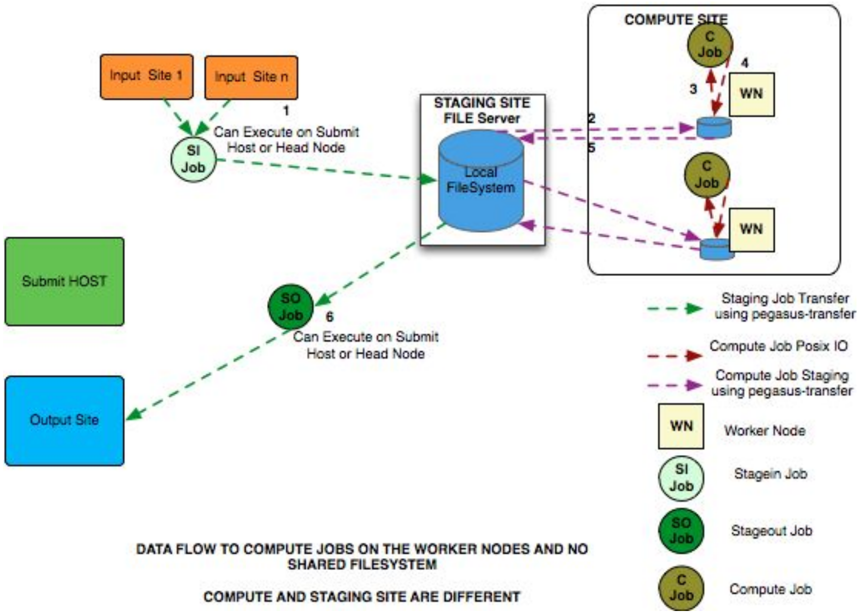
- Worker nodes do not share a file system
- Data is pulled from / pushed to the submit host via HTCondor file transfers
- Staging site is the submit host

Non-shared File System (Clouds, OSG, …)

- Worker nodes do not share a file system
- Data is pulled / pushed from a staging site, possibly not co-located with the computation

Shared File System (HPC sites, XSEDE, Campus clusters, …)

- I/O is directly against the shared file system



Pegasus' internal data transfer tool with support for a number of different protocols

pegasus-transfer

Directory creation, file removal

- If protocol can support it, also used for cleanup

Two stage transfers between incompatible protocols

- e.g., GridFTP to S3 is executed as: GridFTP to local file, local file to S3

Parallel transfers

Automatic retries

Credential management

- Uses the appropriate credential for each site and each protocol (even 3rd party transfers)

```
HTTP
SCP
GridFTP
Globus Online
iRods
Amazon S3
Google Storage
SRM
FDT
Stashcp
Rucio
cp
ln -s
```

Containers are data too!

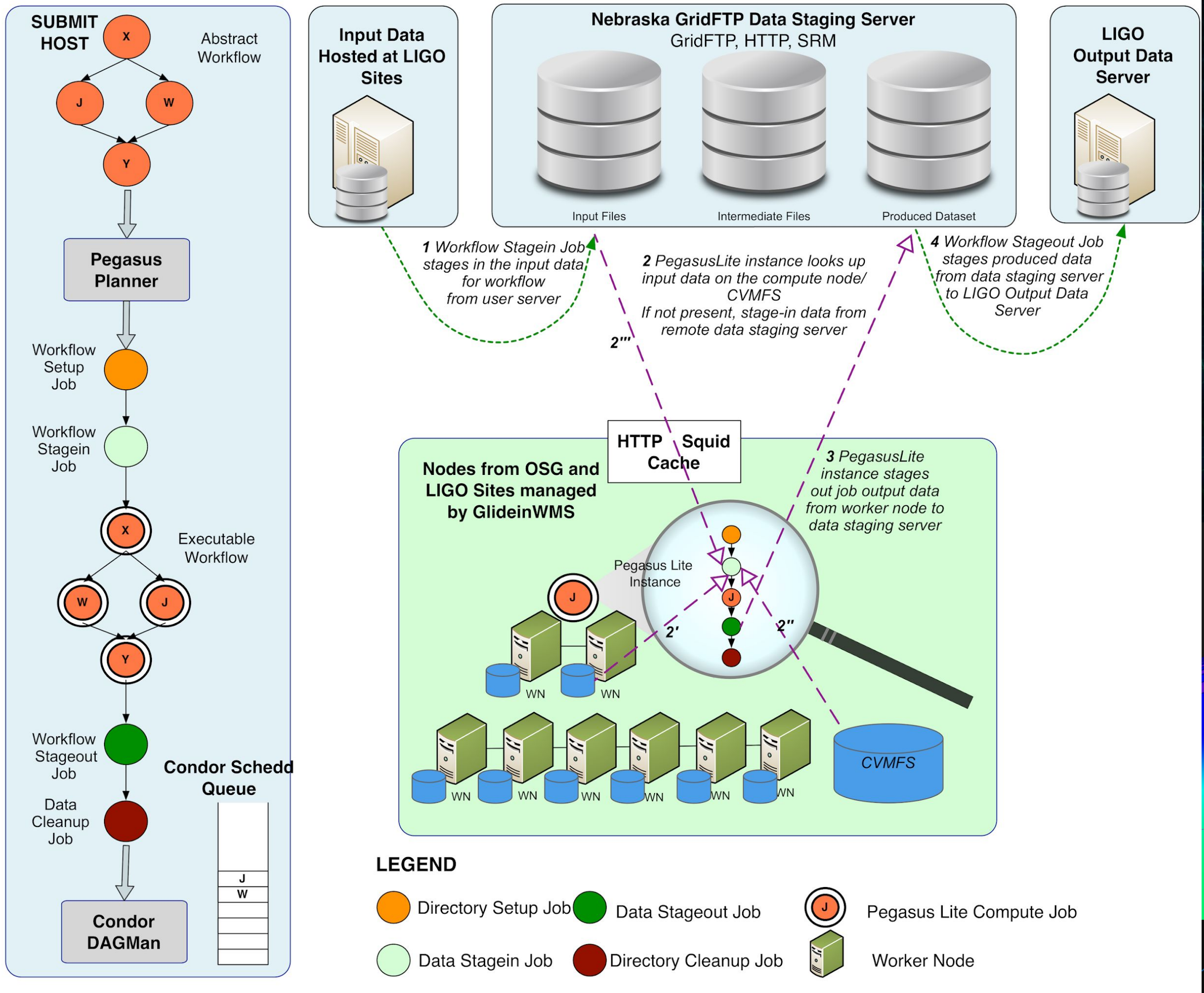
Users can specify to use images from Docker Hub, Singularity Library, or a file using URLs

The image is pulled down as a tar file as part of data stage-in jobs in the workflow

- The exported tar file / image file is then transferred to the job as any other piece of data
- Motivation: Avoid overwhelming Docker Hub/Singularity Library/… with by repeatedly requesting the same image
- Motivation: Optimize workflow data placement and movement

Symlink against a container image if available on shared file systems. For example, CVMFS hosted images on Open Science Grid

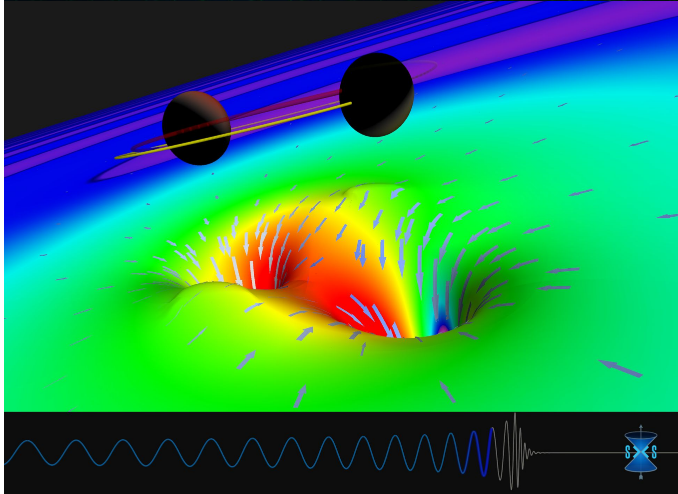
Data Flow for LIGO Pegasus Workflows in OSG



Advanced LIGO –
Laser Interferometer Gravitational Wave Observatory

40,000 compute tasks
Inputs files: 1,100
Output files: 63
Processed Data: 725 GB

Executing on LIGO Data Grid, EGI, Open Science Grid and XSEDE



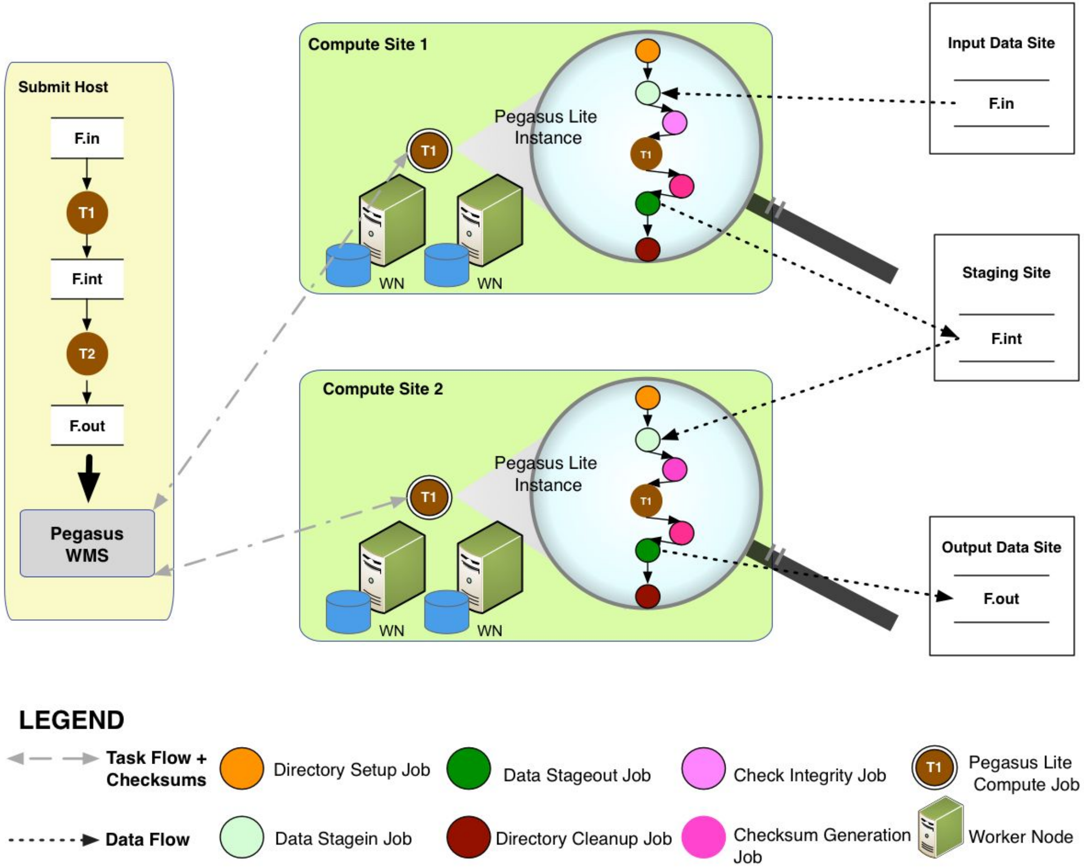
Automatic Integrity Checking

Pegasus performs integrity checksums on input files right before a job starts, ensuring the computation is on the expected piece of data

- For inputs from external sources, checksums specified in the input replica catalog along with file locations, or generated first time we encounter the file

- All intermediate and output files checksums are generated and tracked within the system.

Checksums validation failures is a job failure



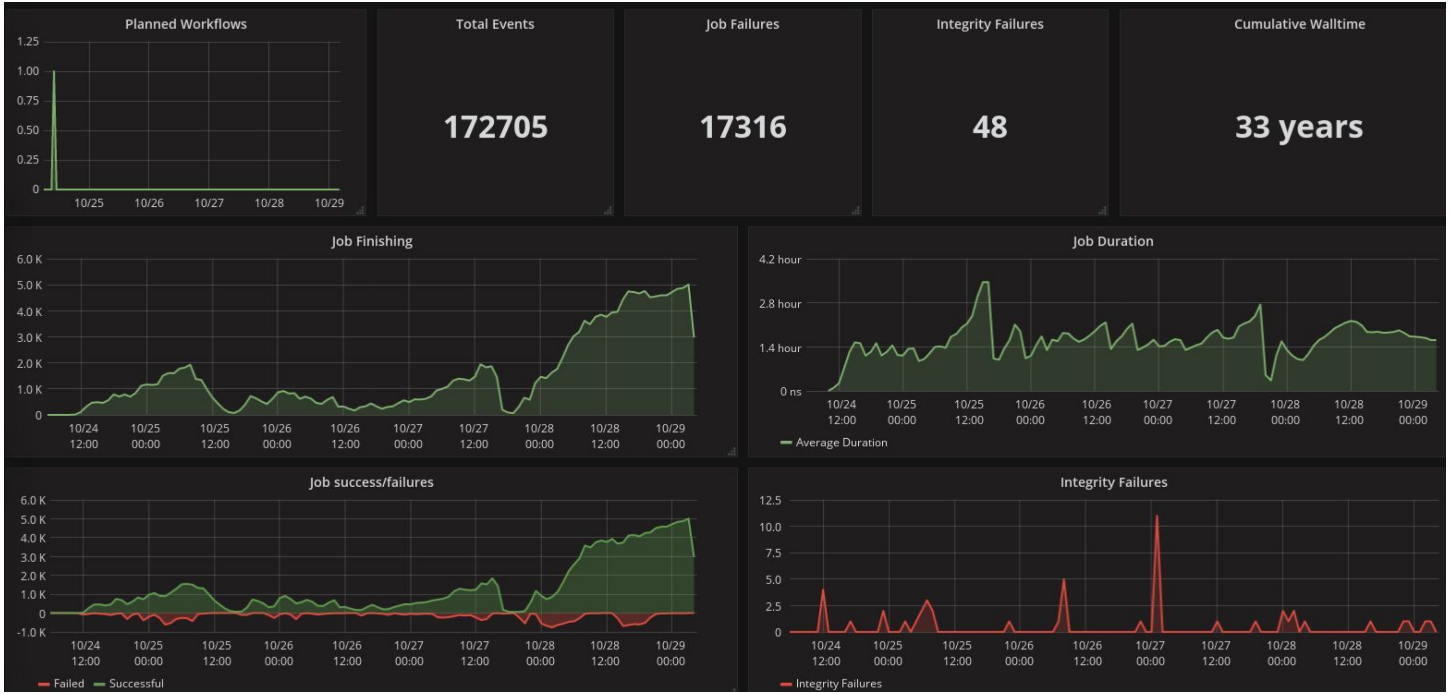
VERITAS / Nepomuk Otte, GATech

Seeing very small, but steady stream of corrected integrity errors from reporting back to Pegasus dashboard.

For VERITAS, ~.04% of transfers fail with integrity errors. (~1 / 2,500 transfers)

Cause uncertain (diagnosis is harder than detection).

Possibly errors in http based transfers (s3 protocol against CEPH)



Pegasus

est. 2001

Automate, recover, and debug scientific computations.

Get Started

Pegasus Website

https://pegasus.isi.edu

Users Mailing List

pegasus-users@isi.edu

Support

pegasus-support@isi.edu

Pegasus Online Office Hours

https://pegasus.isi.edu/blog/online-pegasus-office-hours/

Bi-monthly basis on second Friday of the month, where we address user questions and also apprise the community of new developments

See you at 1PM EST for CI/CS Workshop's Panel: Ups and Downs of Cloud Computing in Open Science