

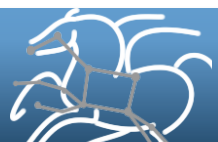
Introduction to Scientific Workflows and Pegasus

Karan Vahi

Science Automation Technologies Group
USC Information Sciences Institute

What is Information Sciences Institute?

- A research organization under the USC Viterbi Engineering. (<http://www.isi.edu>)
 - Birthplace of the internet.
 - 350 fulltime staff. Main Campus in Marina Del Rey.
 - Research Areas
 - Advanced electronics
 - Computational systems and technology
 - Informatics, grid computing , HPC
 - Intelligent Systems – AI , NLP.
- Science Automation Technologies Group (<http://pegasus.isi.edu>)
 - Develops tools and techniques that automate the computational processes.
 - Releases a scientific workflow management system Pegasus.
 - Allows users to run workflows on a variety of infrastructure (local clusters to clouds)
 - Works closely with users to solve their computational problems.
 - Close collaboration with groups at UPC and Keck
 - Southern California Earthquake Center (SCEC - <http://scec.org>)
 - Computational Biology and Bioinformatics (<http://tingchenlab.cmb.usc.edu>)
 - Jim Knowles group at Keck (http://keck.usc.edu/Research/Research_Institutes/Zilkha_Neurogenetic_Institute/Investigators.aspx)

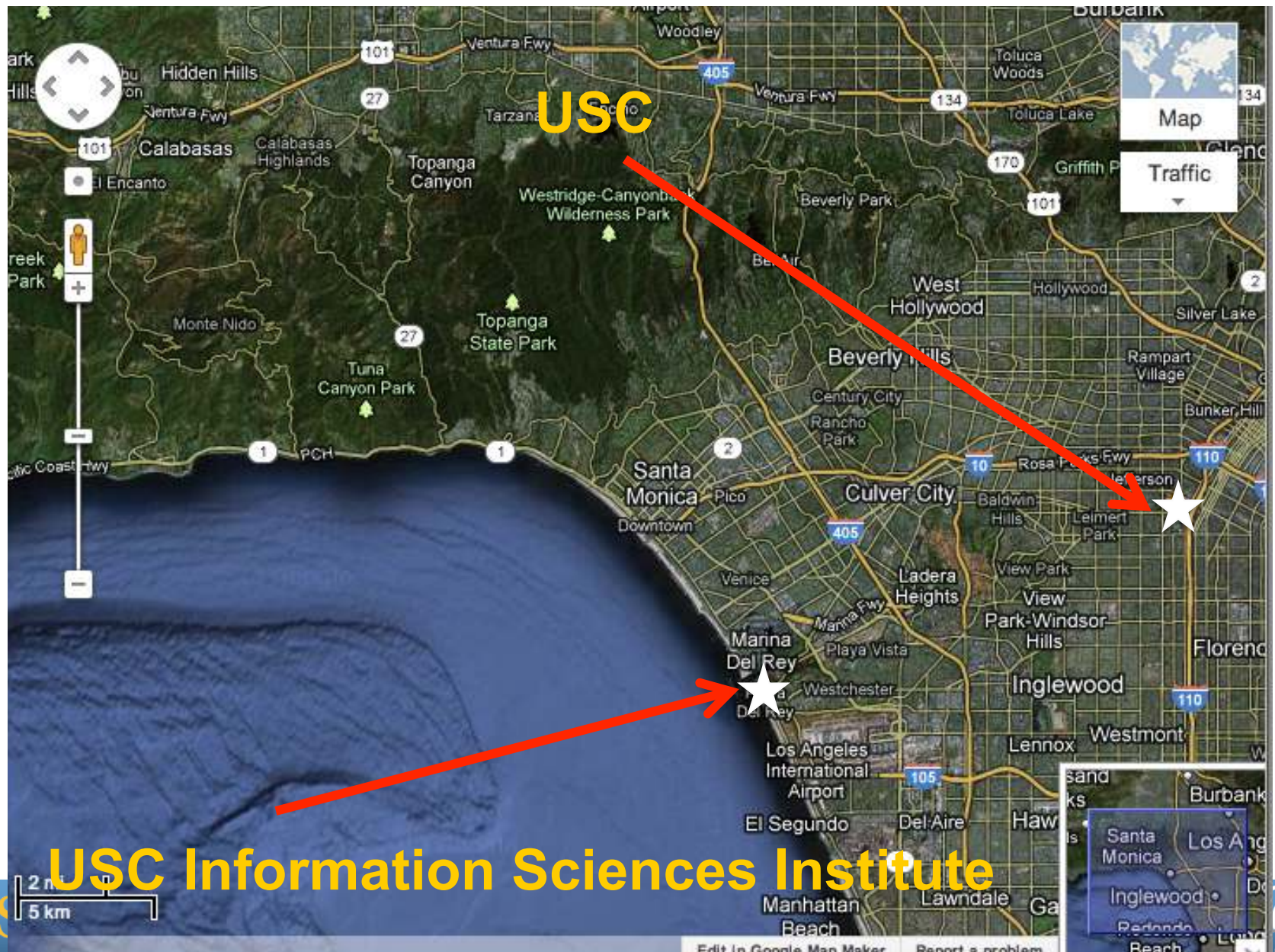




Information Sciences Institute

- Basic and applied research in:
 - Advanced electronics
 - fabrication; novel electronics
 - Computational systems and technology
 - Software/hardware supercomputing, high-performance computing, **cloud computing**, **scientific workflows**
 - Informatics
 - Medical informatics, decision systems, computer networks, grid computing
 - Intelligent systems / artificial intelligence
 - Natural language, knowledge technologies, information
 - and geospatial integration, robotics



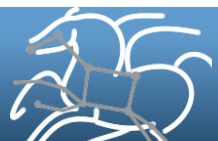


USC

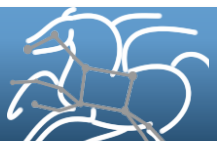
USC Information Sciences Institute

Scientific Workflows

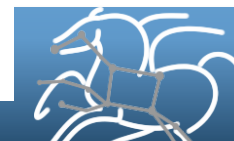
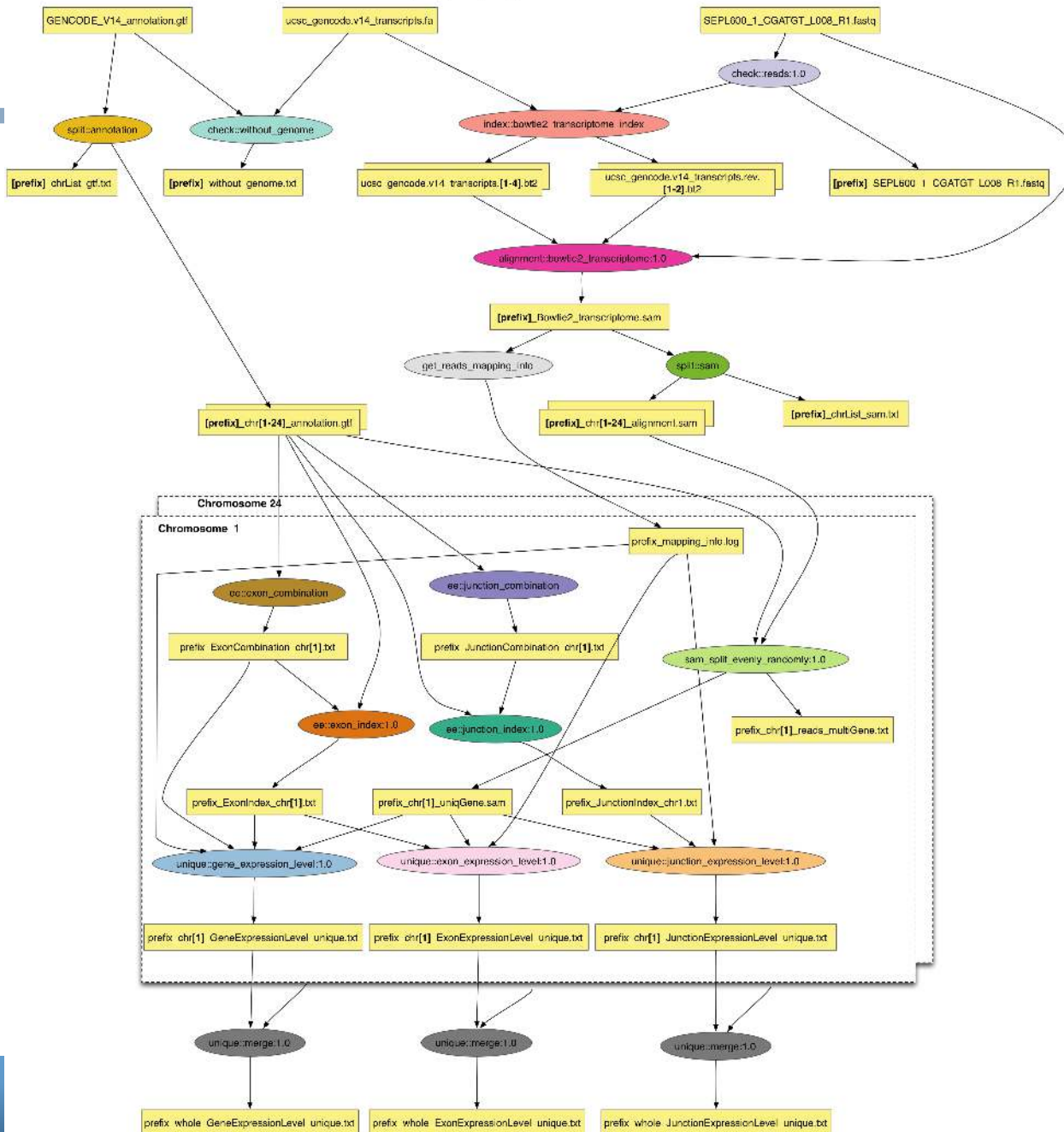
- **Capture individual data transformation and analysis steps**
- **Large monolithic applications broken down to smaller jobs**
 - Smaller jobs can be independent or connected by some control flow/ data flow dependencies
 - Usually expressed as a Directed Acyclic Graph of tasks
- **Allows the scientists to modularize their application**
- **Scaled up execution over several computational resources**
- **Provide automation**
- **Foster Collaborations**



Workflows can be simple!



USC RNASEQ EXPRESSION ESTIMATION WORKFLOW



Computations: Users have same concerns!

■ Data Management

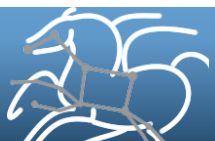
- How do you ship in the small/large amounts data required by your pipeline?
- Different protocols for different sites: Can I use SRM? How about GridFTP? HTTP and Squid proxies?
- Can I use Cloud based storage like S3 on EC2?

■ Debug and Monitor Computations.

- Users need automated tools to go through the log files
- Need to correlate data across lots of log files
- Need to know what host a job ran on and how it was invoked

■ Restructure Pipelines for Improved Performance

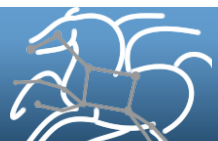
- Short running tasks?
- Data placement?



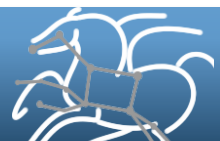
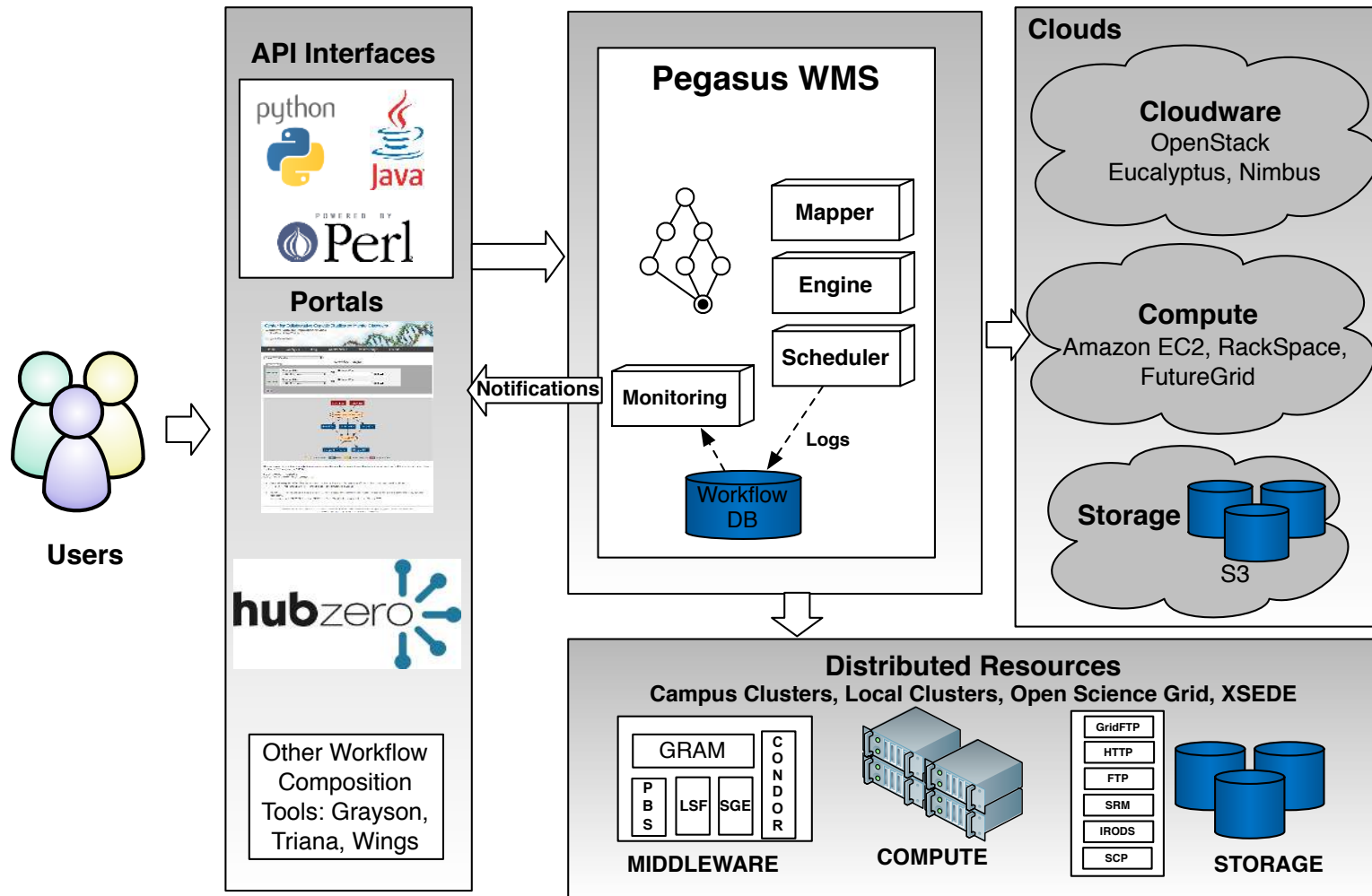
Pegasus

Workflow Management System (est. 2001)

- A collaboration between USC and the Condor Team at UW Madison (includes DAGMan)
- Maps a resource-independent “abstract” workflow onto resources and executes the “executable” workflow
- Used by a number of applications in a variety of domains
- Provides reliability—can retry computations from the point of failure
- Provides scalability—can handle large data and many computations (kbytes-TB of data, $1-10^6$ tasks)
- **Infers data transfers, restructures workflows for performance**
- Automatically captures provenance information
- Can run on resources distributed among institutions, laptop, campus cluster, Grid, Cloud



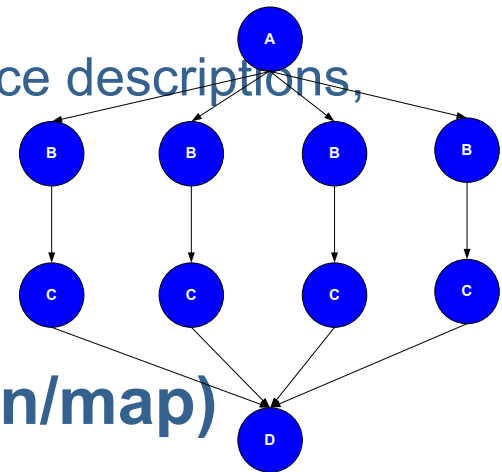
Pegasus WMS



Pegasus Workflow Management System

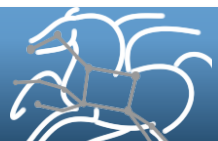
■ Abstract Workflows - Pegasus input workflow description

- Workflow “high-level language”
- Only identifies the computation, devoid of resource descriptions, devoid of data locations
- File Aware



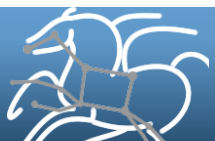
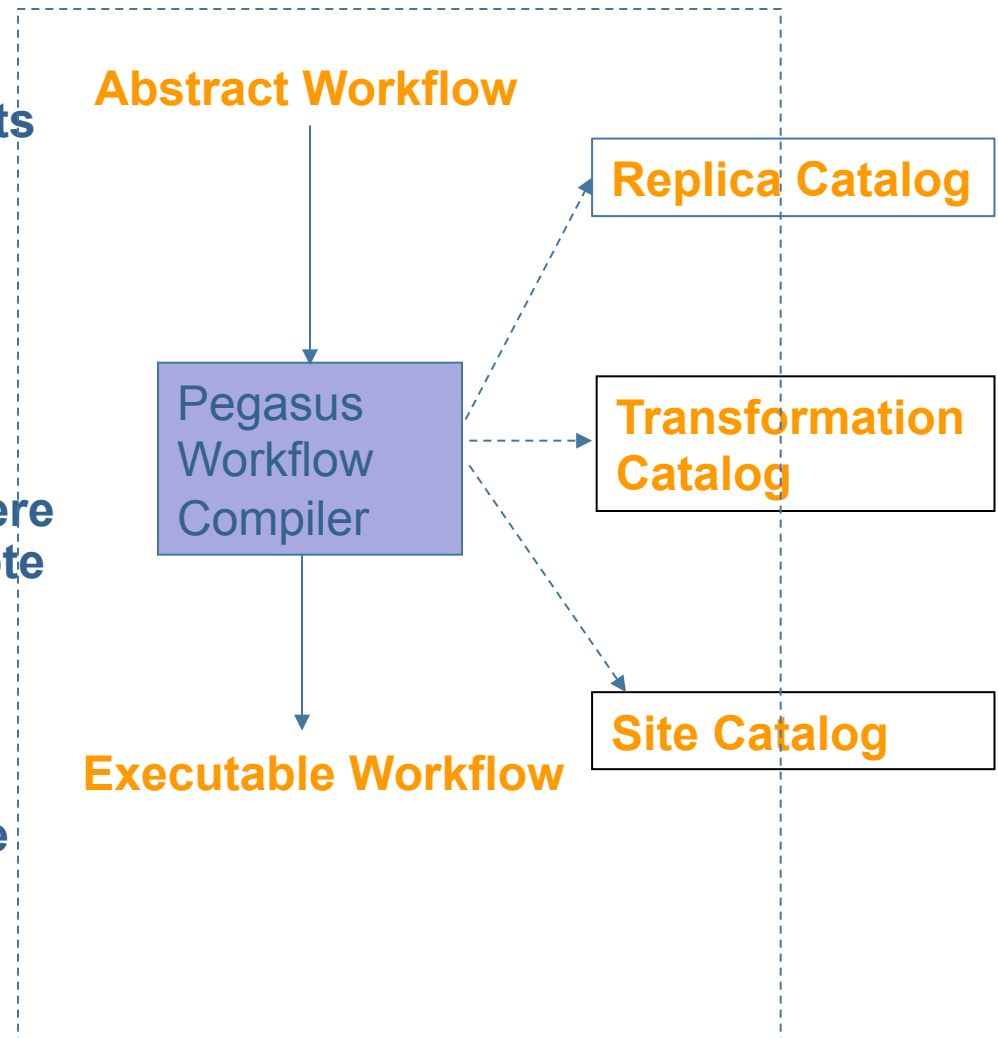
■ Pegasus is a workflow “compiler” (plan/map)

- Target is DAGMan DAGs and Condor submit files
- Transforms the workflow for performance and reliability
- Automatically locates physical locations for both workflow components and data
- Collects runtime provenance

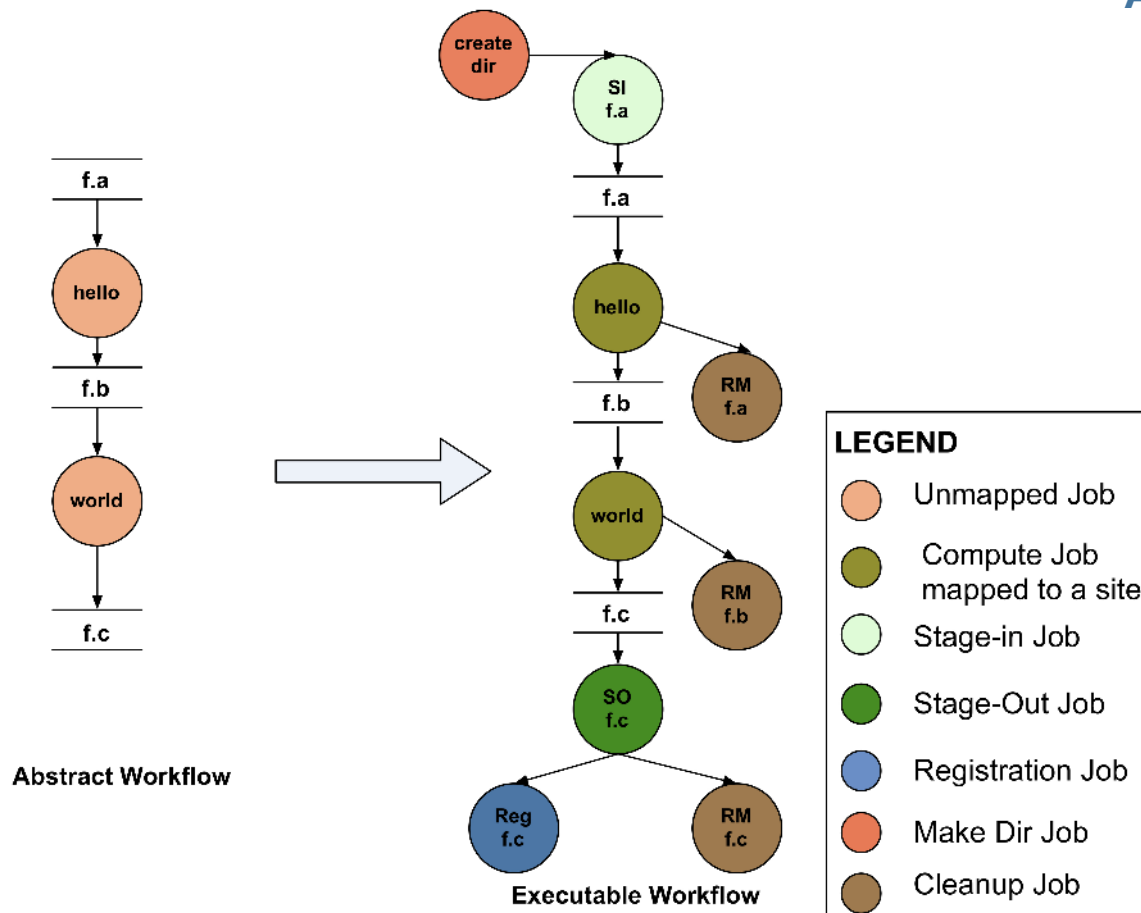


Abstract to Executable Workflow Mapping - Discovery

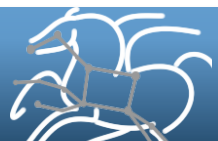
- **Data**
 - Where do the input datasets reside?
- **Executables**
 - Where are the executables installed ?
 - Do binaries exist somewhere that can be staged to remote grid sites?
- **Site Layout**
 - What does a execution site look like?



Abstract to Executable Workflow Mapping



- **Abstraction provides**
 - **Ease of Use** (do not need to worry about low-level execution details)
 - **Portability** (can use the same workflow description to run on a number of resources and/or across them)
 - **Gives opportunities for optimization and fault tolerance**
 - automatically restructure the workflow
 - automatically provide fault recovery (retry, choose different resource)



What Does Pegasus provide an Application - I

- **Portability / Reuse**

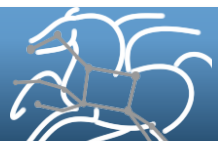
- User created workflows can easily be mapped to and run in different environments without alteration.

- **Data Management**

- Pegasus handles replica selection, data transfers and output registrations in data catalogs. These tasks are added to a workflow as auxiliary jobs by the Pegasus planner.

- **Performance**

- The Pegasus mapper can reorder, group, and prioritize tasks in order to increase the overall workflow performance.



What Does Pegasus provide an Application - II

■ Provenance

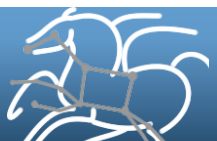
- Provenance data is collected in a database, and the data can be summaries with tools such as pegasus-statistics, pegasus-plots, or directly with SQL queries.

■ Reliability and Debugging Tools

- Jobs and data transfers are automatically retried in case of failures. Debugging tools such as pegasus-analyzer helps the user to debug the workflow in case of non-recoverable failures.

■ Scalability

- Hierarchal workflows
- Scale to hundreds of thousands of nodes in a workflow.



Simple Steps to Run Pegasus

1. Specify your computation in terms of DAX

- Write a simple DAX generator
- Python, Java , Perl based API provided with Pegasus

2. Set up your catalogs

- Replica catalog, transformation catalog and site catalog.

3. Plan and Submit your workflow

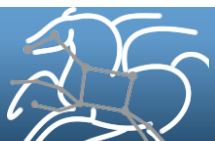
- Use *pegasus-plan* to generate your executable workflow that is mapped onto the target resources and submits it for execution

4. Monitor and Analyze your workflow

- Use *pegasus-status* | *pegasus-analyzer* to monitor the execution of your workflow

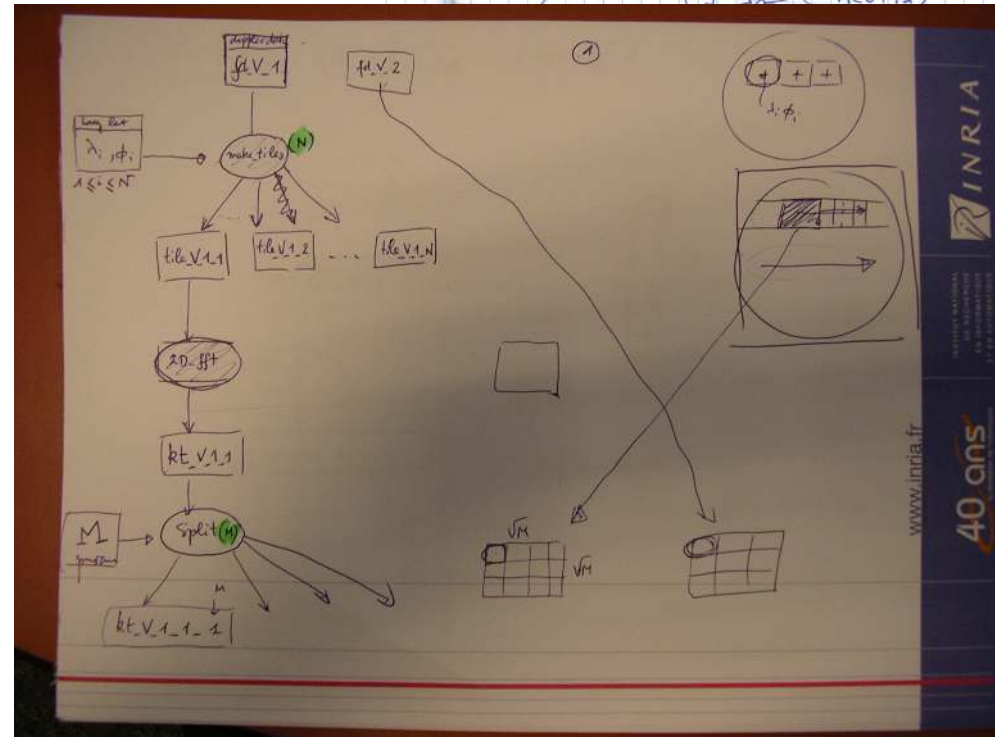
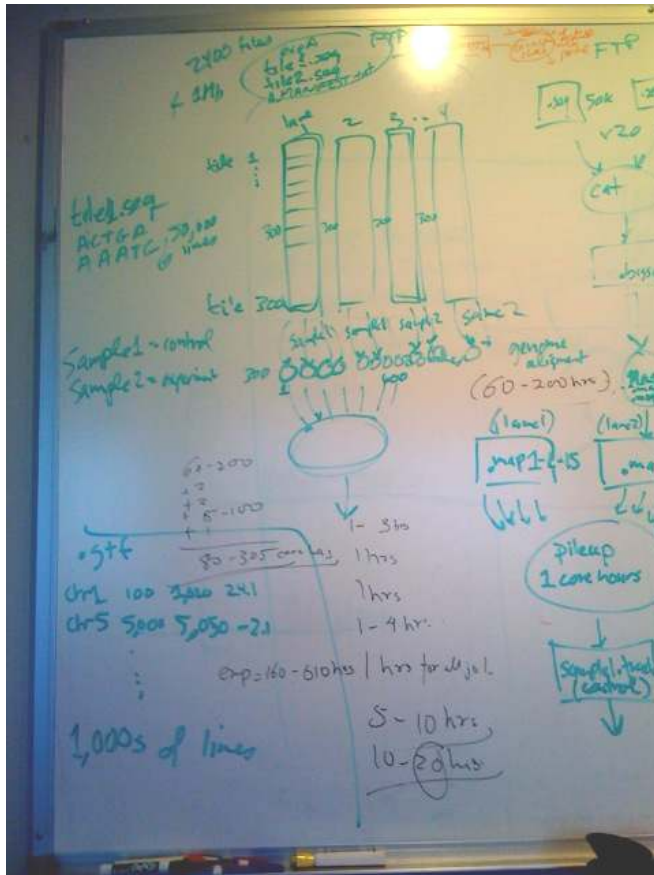
5. Workflow Statistics

- Run *pegasus-statistics* to generate statistics about your workflow run.



If you get stuck...

And you can draw....



We can help you!

Support: pegasus-support@isi.edu
pegasus-users@isi.edu



Relevant Links

- Pegasus: <http://pegasus.isi.edu>
- Tutorial and documentation:
<http://pegasus.isi.edu/wms/docs/latest/>
- Support: pegasus-users@isi.edu
pegasus-support@isi.edu

Acknowledgements

Pegasus Team, Condor Team, funding agencies, NSF, NIH, and everybody who uses Pegasus.

