Origami: Scientific Distributed Workflow in McIDAS-V

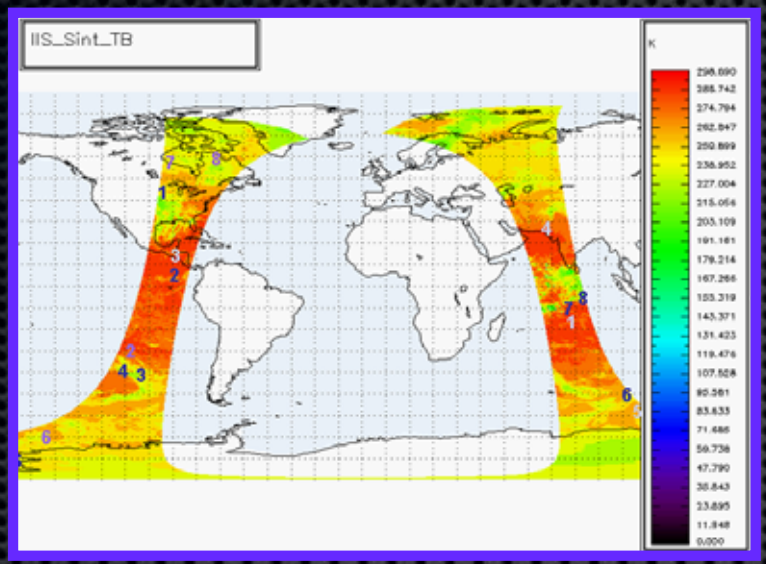
Maciek Smuga-Otto, Bruce Flynn
(also Bob Knuteson, Ray Garcia)
SSEC

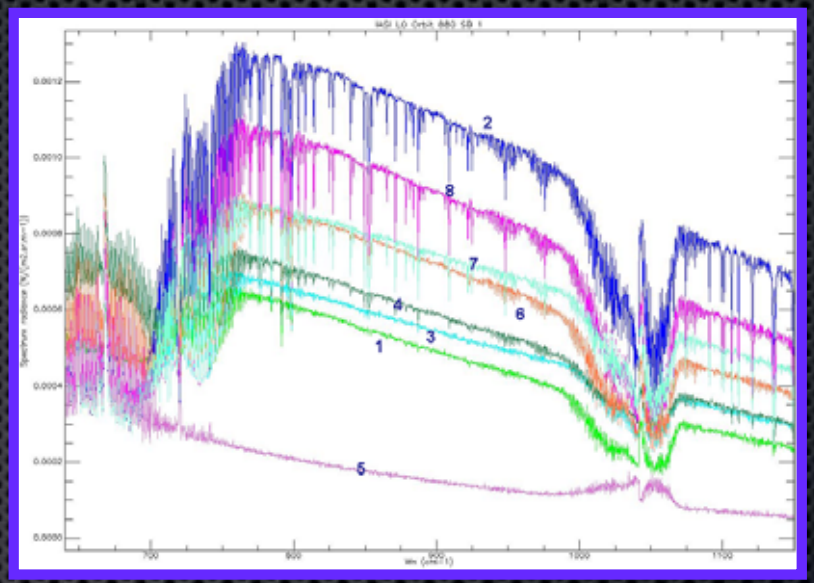
1. Motivation and Concepts

- new remote sensing instruments
- new computing resources
- need for science workflow
- dealing with data

SSEC

New Remote Sensing Instruments





- higher spatial, temporal, spectral resolutions
- enormous data volume
- complex calibration procedures
- Example: IASI hyperspectral sounder

SSEC

New Computing Resources

- compute clusters
- distributed storage of large data volumes
  - SAN (Storage Area Network)
  - cluster file systems (such as LUSTRE)
  - SRB (Storage Resource Broker)
- emerging computational grids

Dealing with Data

- Desired actions with data:
  - search for data by time/location
  - search for data by feature
  - share results with community
- data volumes too unwieldy to store locally

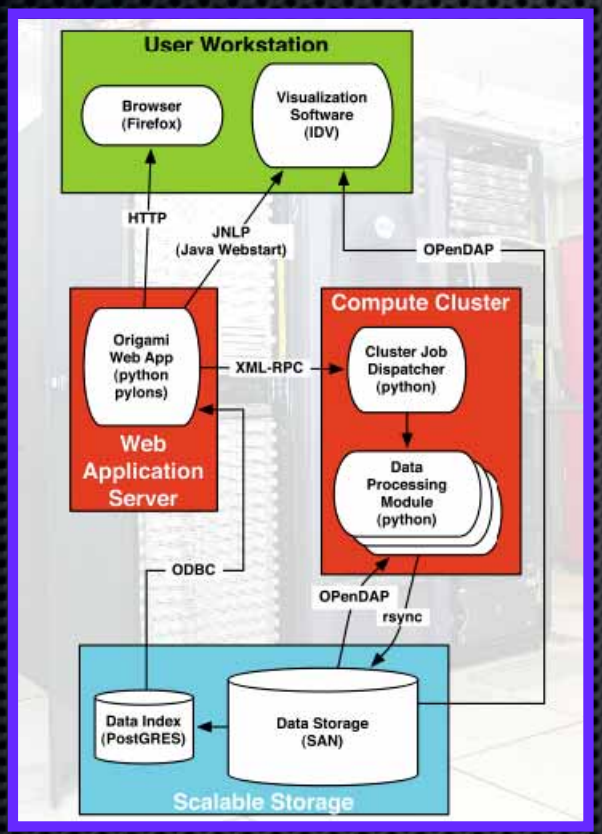
"QuickTime™ and a GIF decompressor are needed to see this picture."

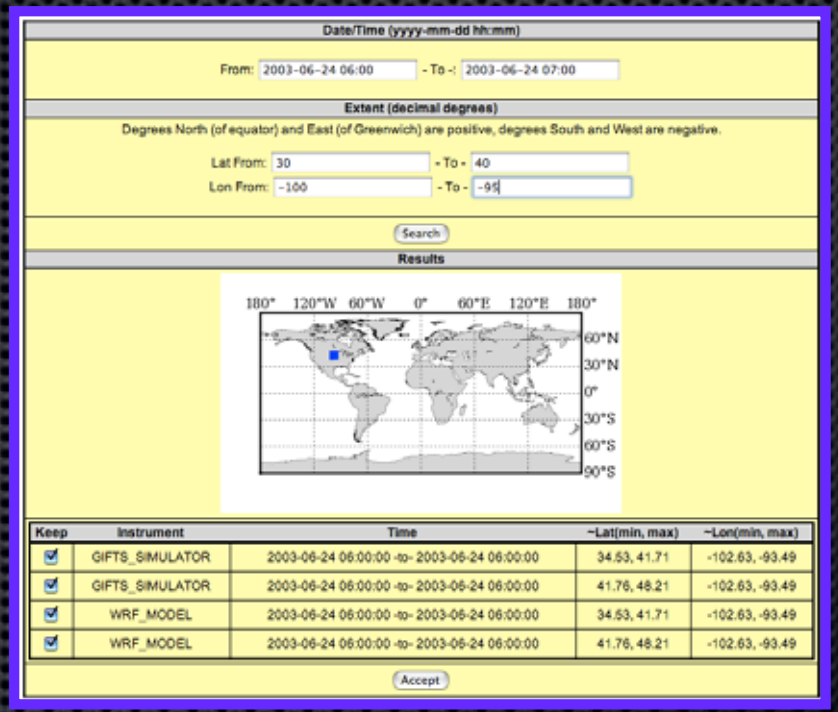
SSEC

Need for Science Workflow Management

- solving the same problem over and over
  - search for relevant data
  - run time-intensive computation (perhaps in parallel)
  - collect/visualize results
  - repeat with different data/parameters
  - publish

First Impulse: “Do it all Yourself”

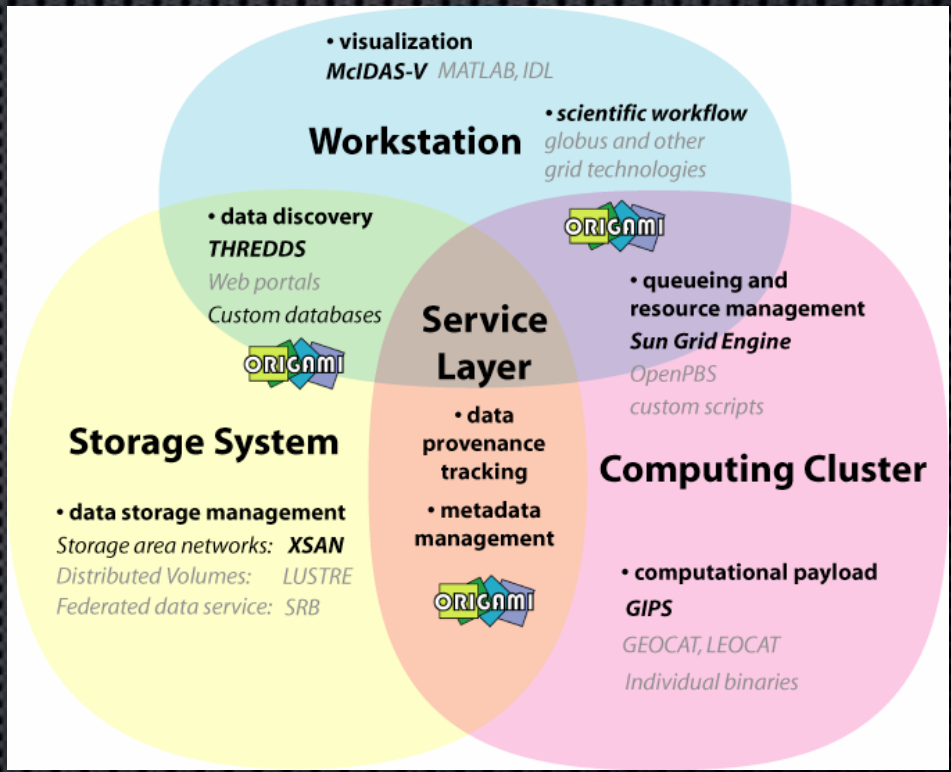




SSEC

A New Architecture

- integrate existing components
- build where necessary
- glue with flexible scripts

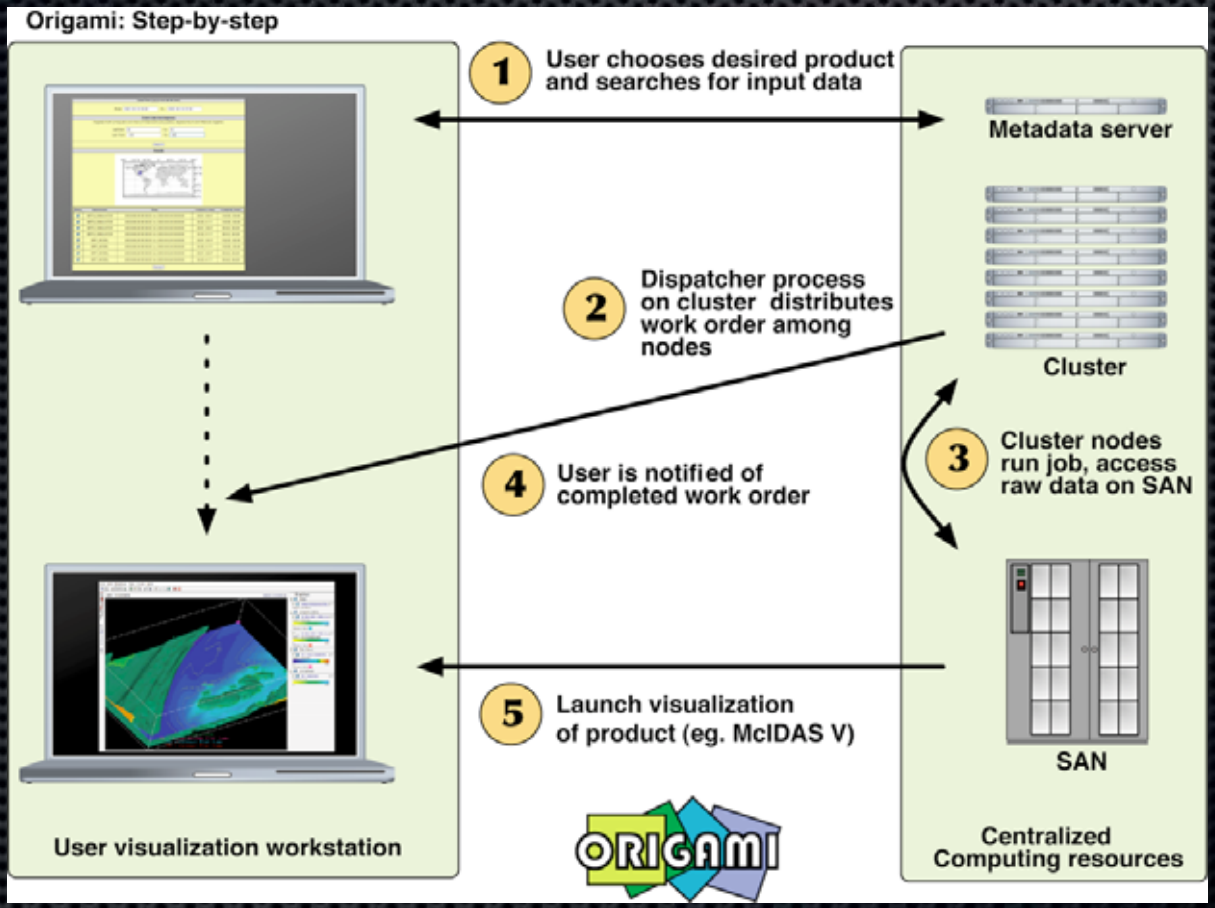


SSEC

2. Origami

- use lightweight scripting environment (python)
  - to prototype distributed scientific workflow
  - make capabilities accessible from McIDAS-V
- manage the workflow relying on existing tools
  - develop metadata standards as “glue”
  - develop tools to manipulate and use the metadata

The Origami Workflow



SSEC

McIDAS-V and Origami

- Part of the power of McIDAS-V lies in the external resources it can harness:
  - access to large remote data volumes
  - access to remote computation farms
- potential to control this environment from the desktop

Demo of Origami on McIDAS-V

- today: demonstrating mockup of functionality
  - concentrating on McIDAS-V integration
- previous web interface version ran a simple example algorithm to calculate relative humidity from T WV fields.

Step 0: Register Algorithm

- user registers algorithm with system
  - either a standalone executable
  - or as a library within a deployment framework (GEOCAT, LEOCAT)
  - or as python source code relying on common numerical/science libraries
- attach description of algorithm interface in XML format

Step 1: Get Data

- user selects registered algorithm, searches for data
  - data search constrained by algorithm interface (so irrelevant data are automatically excluded)
  - currently, only search by time/location
  - interactive interface: “rubber band box selection”
- can also browse data manually

Step 1a: Create Work Order

- user specifies how the job is to run (which data on what executables)
- McIDAS-V front-end to a generic web service
  - can be accessed by other applications
  - or can submit a XML description file

Steps 2, 3: Compute!

- in background:
  - 2. job is dispatched to compute nodes
  - 3. data is delivered to nodes as needed

Step 4: Track Progress

- user queries progress of jobs
  - system notifies user of completed job
  - user can browse description of current and past jobs

Step 5: Gather data, Visualize, Etc.

- system routes product to desktop visualization environment
  - using OPeNDAP for the data
- registers product along with metadata for future searches and downstream processing

Questions, comments

- Team: Bob Knuteson, Ray Garcia, Bruce Flynn, Maciek Smuga-Otto
- contact: maciek@smuga-otto.com
- Thanks!