Simplified, Parallelized InSAR Scientific Computing Environment

Paul A. Rosen, Eric Gurrola, Piyush Agram,
Geoffrey Gunter
Jet Propulsion Laboratory
California Institute of Technology

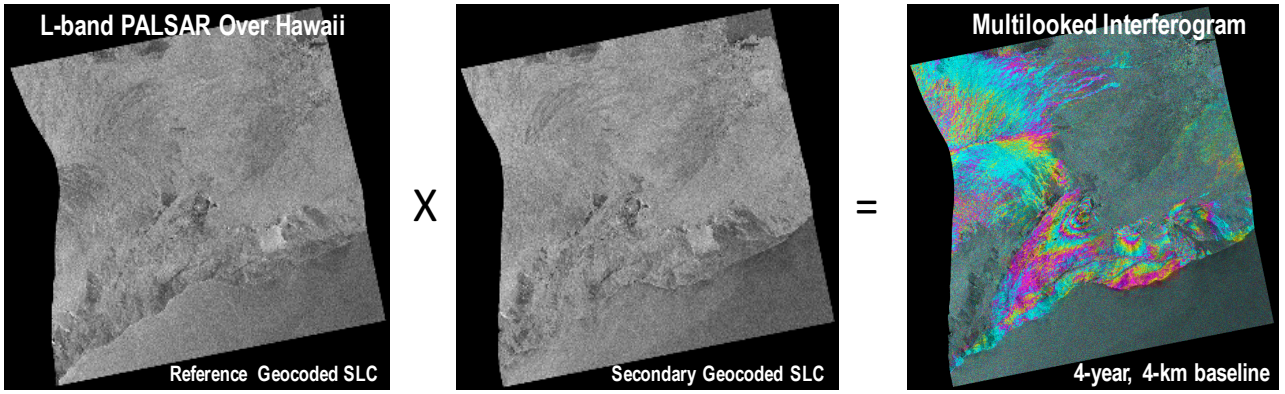
Michael Aivazis
Parasim, Inc.

Howard Zebker
Stanford University

Earth Science Technology Forum
NASA Ames Conference Center
June 12, 2019

Simplified Parallelized InSAR Scientific Computing Environment (SPISCE) Background

- Past ESTO investments have created the InSAR Scientific Computing Environment (ISCE), used by individual users and semi-operational production systems (ARIA)
- The growth in data from Sentinel 1 and soon NISAR requires a fundamentally different way of interacting with data
- This work exploits the power for GPUs to slice through data and implement framework elements to give scientists a different user experience



GPU-accelerated Geocoded SLC data and related workflows will help users create derived products with simple editable operations

Objectives

- Develop a workflow that generates single look complex (SLC) images in an easily ingestible form for data utility, incorporating all necessary InSAR phase corrections and resampling to provide products that anyone can understand
- Develop methods of computational acceleration by exploiting back projection methods on cloud-enabled GPU platforms to directly compute focused and InSAR-corrected imagery in UTM (Landsat grid).
- GPU-parallelize and otherwise accelerate key algorithms in InSAR Scientific Computing Environment (ISCE) to gain factors of 10-100 increase in speed.
- Develop framework technologies at the user interface that support a more natural way for scientists to specify products and actions.
- Provide open source solutions

Summary of Results

Geocoded SLC workflow

- Developed the workflow – foundation of data processing in Stanford's research group
- InSAR process flow experimentation
  - Is it needed to create redundant interferograms to get best results in the context of the starting point of geocoded SLC?
- Socializing geocoded SLC's with the community
  - Papers at AGU
  - Papers at LPS and IGARSS this year.

GPU parallelization of back-projection processor

- Significant (10-100 X) improvements to speed and efficiency – 2 seconds per image

GPU parallelization of ampcor processor

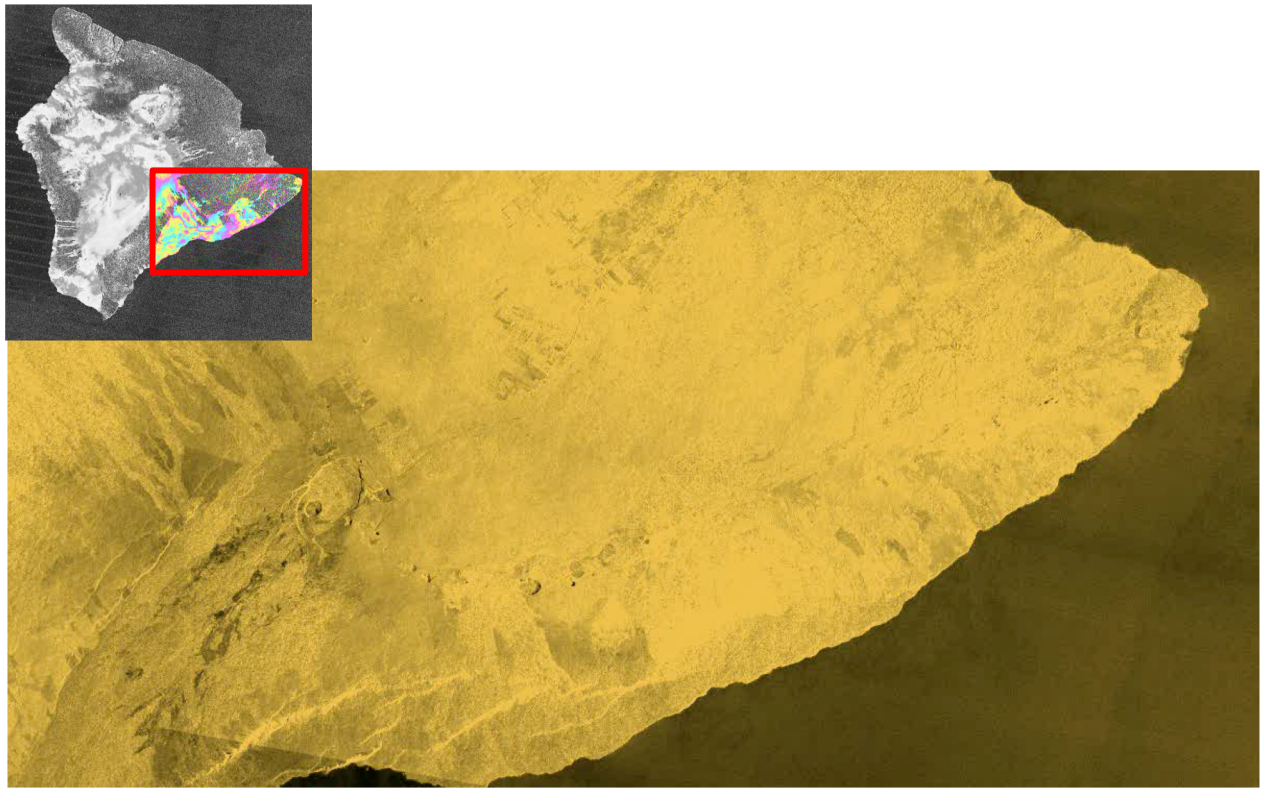
- Excellent (1000 X) improvement in speed and efficiency

Pyreized workflow

- ISCE 2 components upgrade to pyre – "complete"
- SPISCE 1.0 with GUI showing each component as a workflow node near complete

User-friendly products

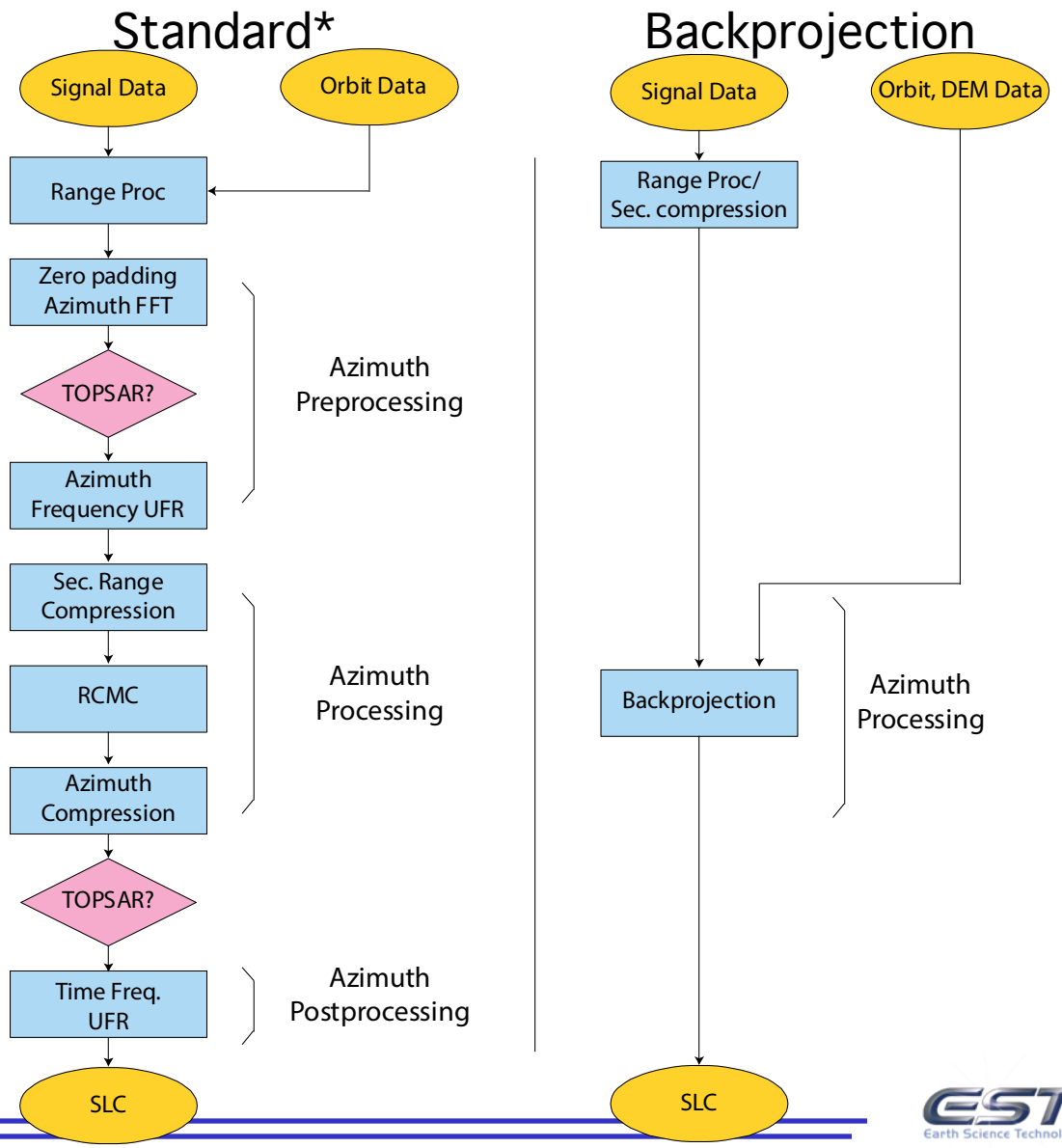
- Creating a time series view of Kilauea eruption is greatly simplified by starting with geocoded SLCs facilitated by backprojection



Traditional "range-doppler" Image – hard to interpret!

L0 SLC processing for TOPS

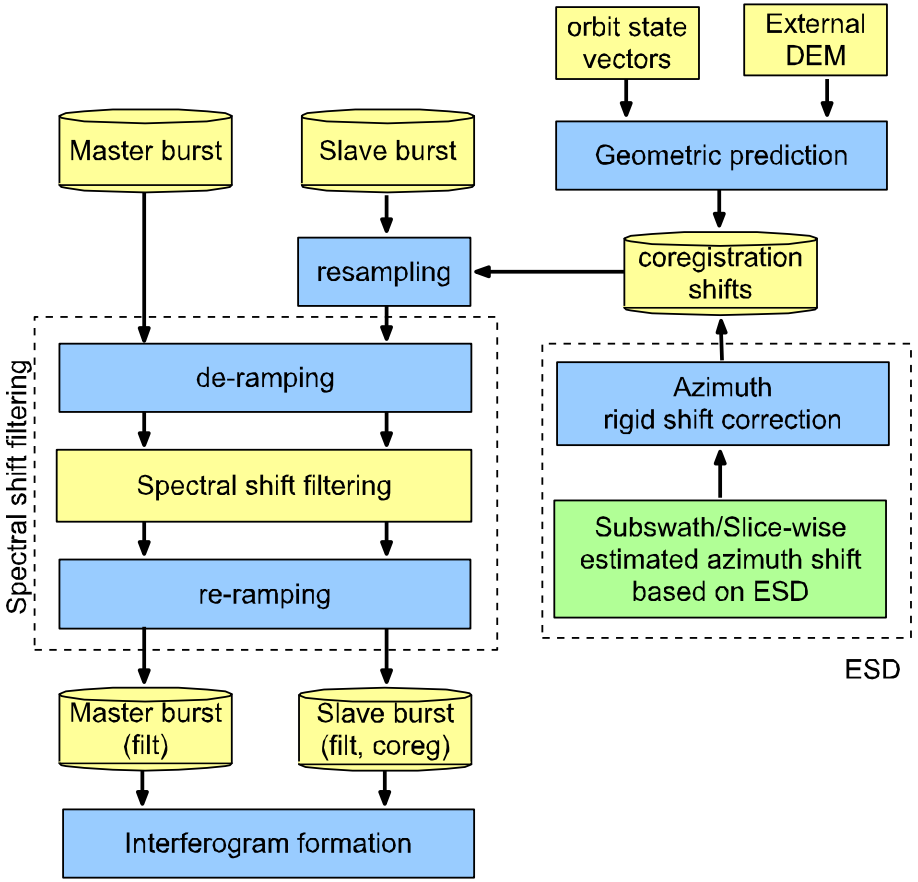
- Dealiasing computationally expensive and very intricate though elegant
- Backprojection same for all modes
- Short integration time makes backprojection efficient



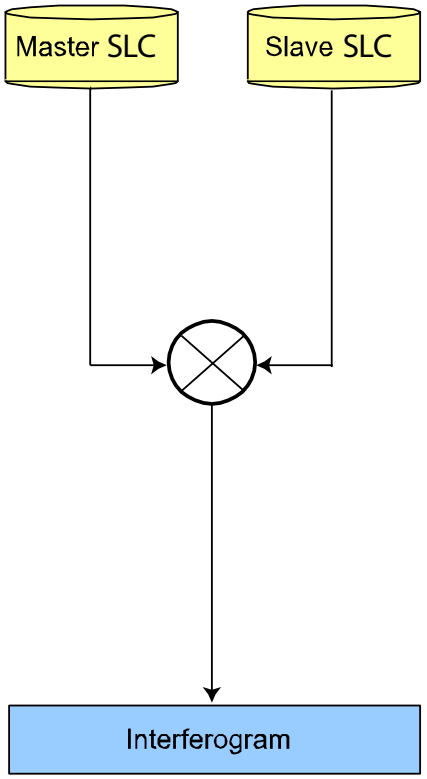
*Sentinel 1 SAR Technical Guide

Interferogram formation – by user

Standard (Yague-Martinez et al.,2016)



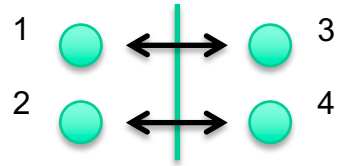
Backprojection



Example: Noise decreases, accuracy increases with "redundant" data generation

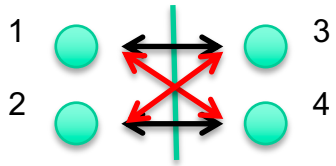
$$\phi_{ij} = \phi_{ij}^{defo} + \phi_{ij}^{aps} + \phi_{ij}^{decor}$$

Independent stacking



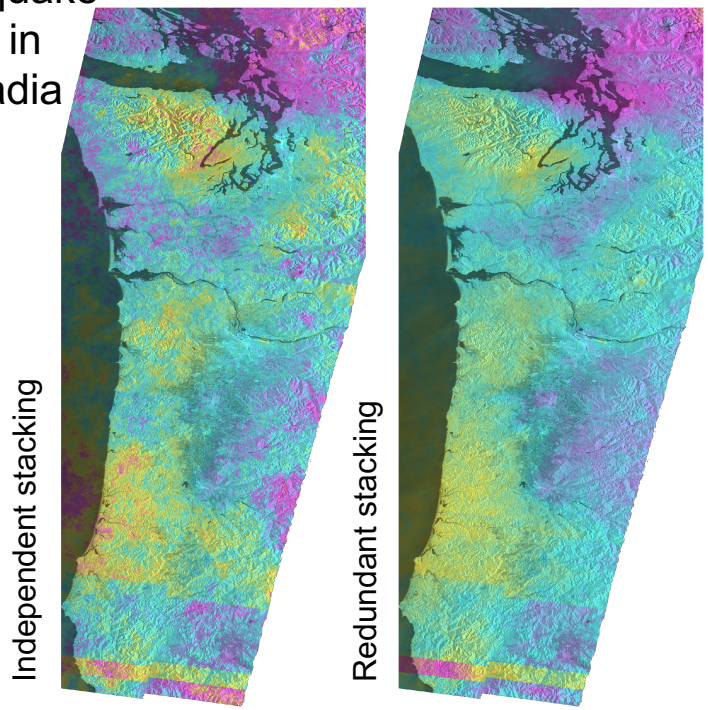
$$\phi_{ind} = \frac{1}{2}(\phi_{13} + \phi_{24})$$

Redundant stacking



$$\phi_{red} = \frac{1}{4}(\phi_{13} + \phi_{24} + \phi_{14} + \phi_{23})$$

Slow slip earthquake signal in Cascadia



Can now generate "redundant" interferograms where before it was not practical

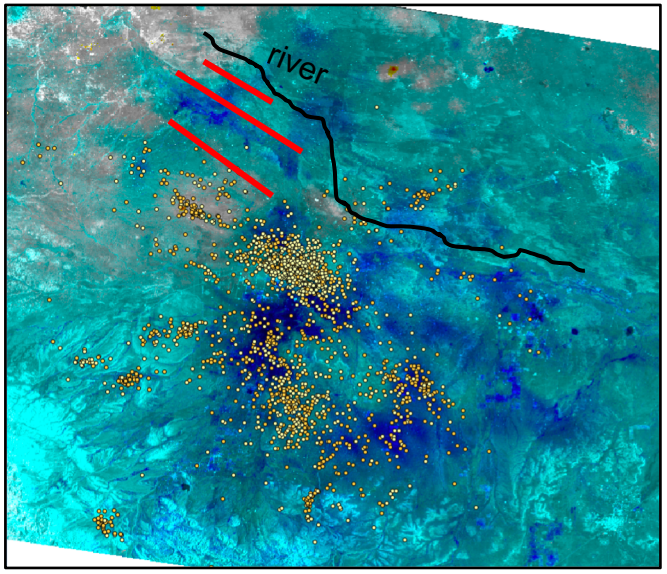
Conclusion: *AIST improved workflow makes InSAR sensitive and accurate for many more events and studies*

Example of Socialization of Products with Community

Observed Deformation and Seismicity Application
From 1407 interferograms enabled by AIST backprojection processor

Induced seismicity from oil and gas production – Delaware Basin, Texas

Participants: Texas Bureau of Economic Geology, Stanford Center for Induced and Triggered Seismicity

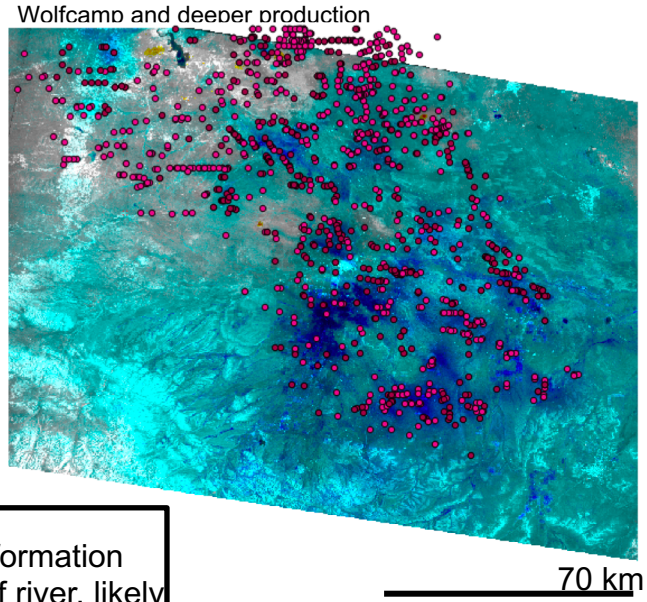


<----- subsidence uplift -----> 40 km

-10 -8 -6 -4 -2 0 2 4 6 8 10
Weighted LOS Displacement (cm)

At left: Seismicity (eq's) correlates well with observed deformation signal

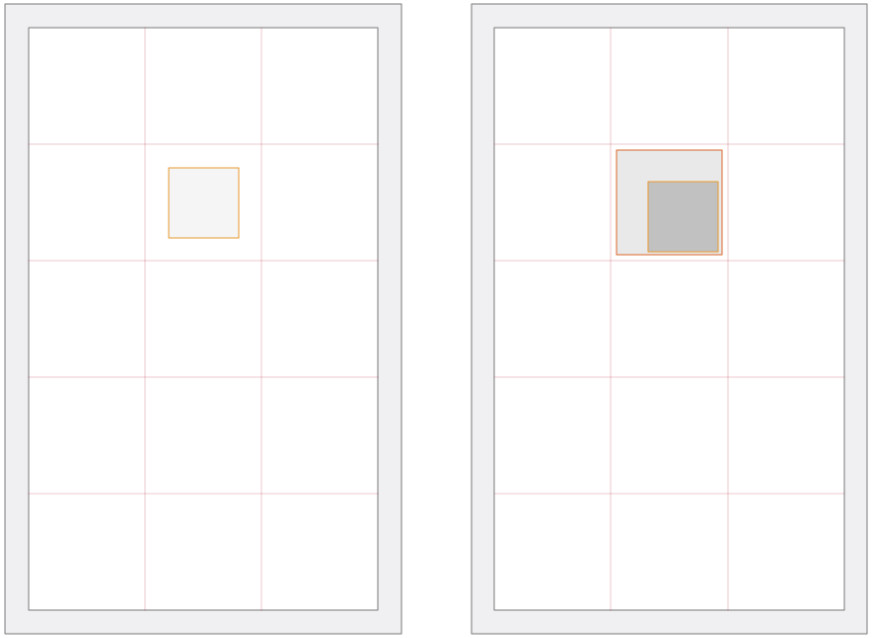
At right: Production correlates well with observed deformation signal south of river



Neither seismicity nor deformation correlate spatially north of river, likely due to geological setting and fault type

Image cross-correlation GPU-acceleration

- Slide a tile within the search window whose shape is the same as the reference tile
- Compute a correlation measure at each possible placement
- Record the placements of maximum correlation for each tile
- Build and post-condition a correlation surface to form the offset field



reference image

target image

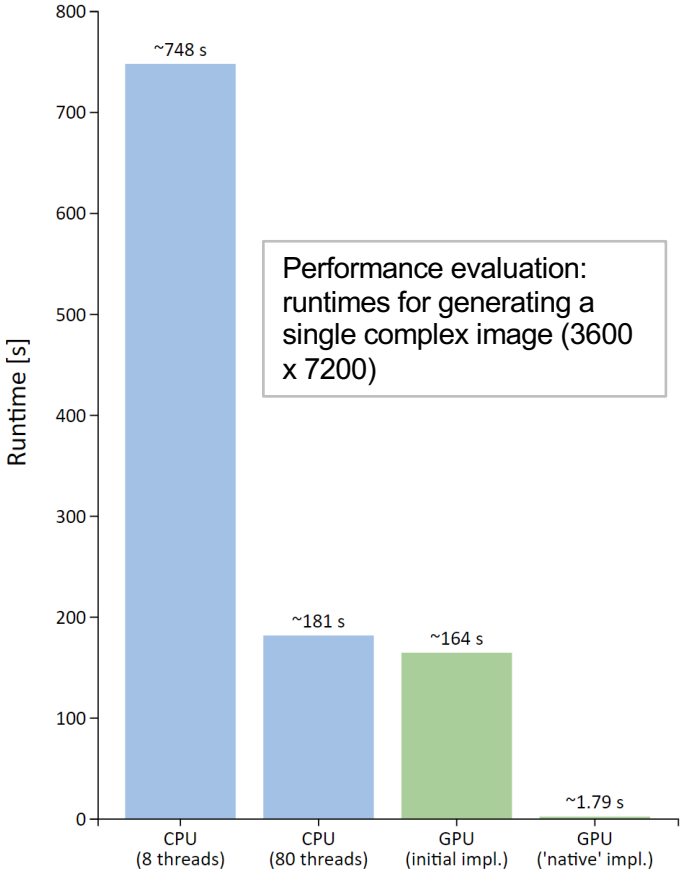
C++ intermediary code improvements (20X)

- memory mapped IO
- algorithms from the C++ STL
- a dedicated data structure to simplify multi-dimensional indices arithmetic

GPU code improvements (addl. 1500X)

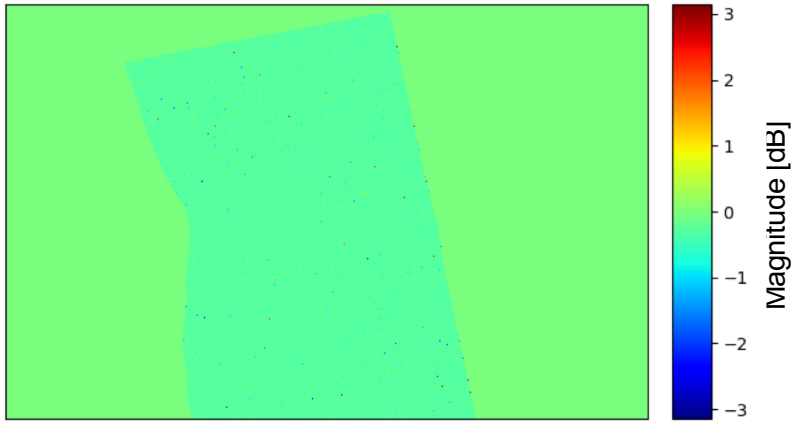
- careful packing of the input tiles in memory
- minimizing the data transfer cost between host and device
- eliminating memory bank conflicts within the device
- highly tuned implementation of the reduction kernels

GPU-Accelerated back projection comparison

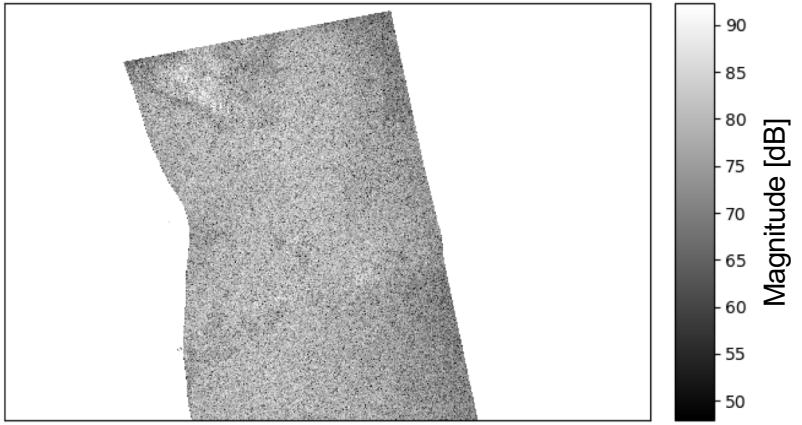


Performance evaluation: runtimes for generating a single complex image (3600 x 7200)

CPU: Intel Xeon CPU E5-2698 v4 | GPU: Nvidia Tesla V100

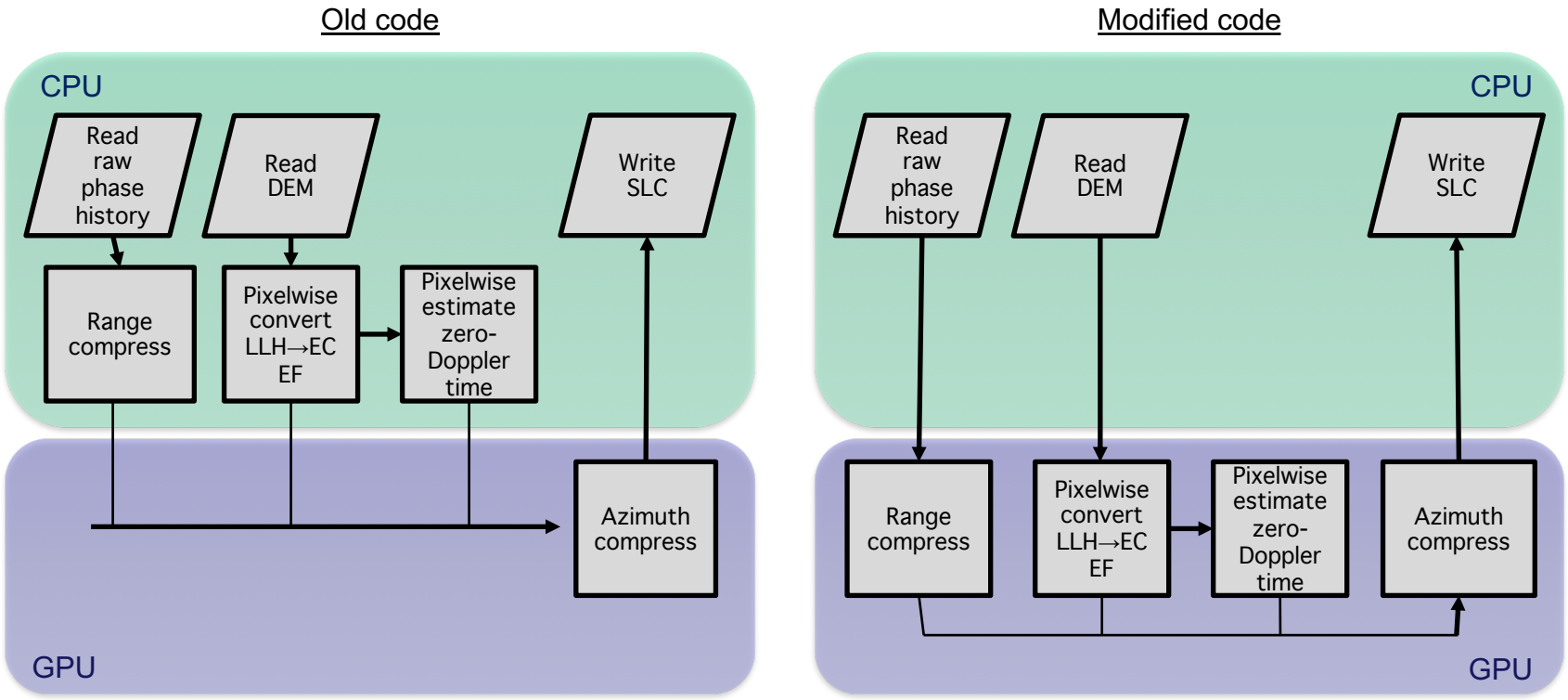


Interferometric phase between CPU & GPU SLCs



GPU ('native' impl.)

GPU backprojection code modifications



Workflow visualizer (Flo)

- We are building a web-based application
  - simplify and organize access to the dozens of configurable parameters
  - support interactive processing
  - embedded within the isce application
- Managing the user experience for such a complex system is a challenge
  - multiple layers of information
  - complicated workflows tend to become difficult to render in an understandable way
  - users have high expectations from such interfaces
- Dozens of attempts to solve this problem
  - we think we have a good chance to avoid the known pitfalls, and provide a workable solution

Visual language prototype

atomic data

composite data

composite data with detail

nameless factory

named factory

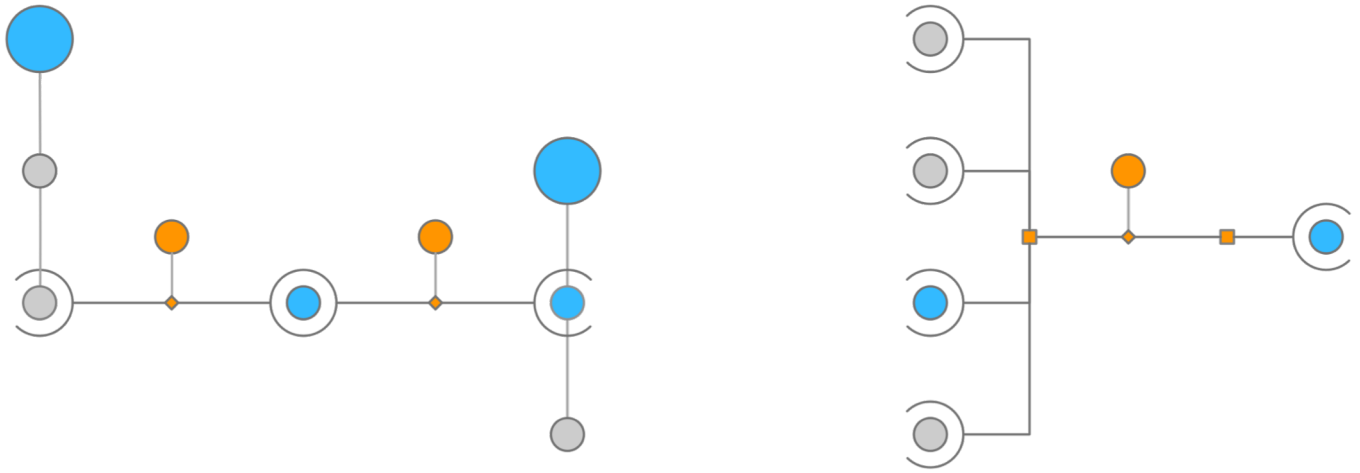
n x m factory

raw input

final product

data bindings

Simple flow examples



Server:

- isce+pyre (GitHub: isce-framework/isce2 and pyre/pyre)

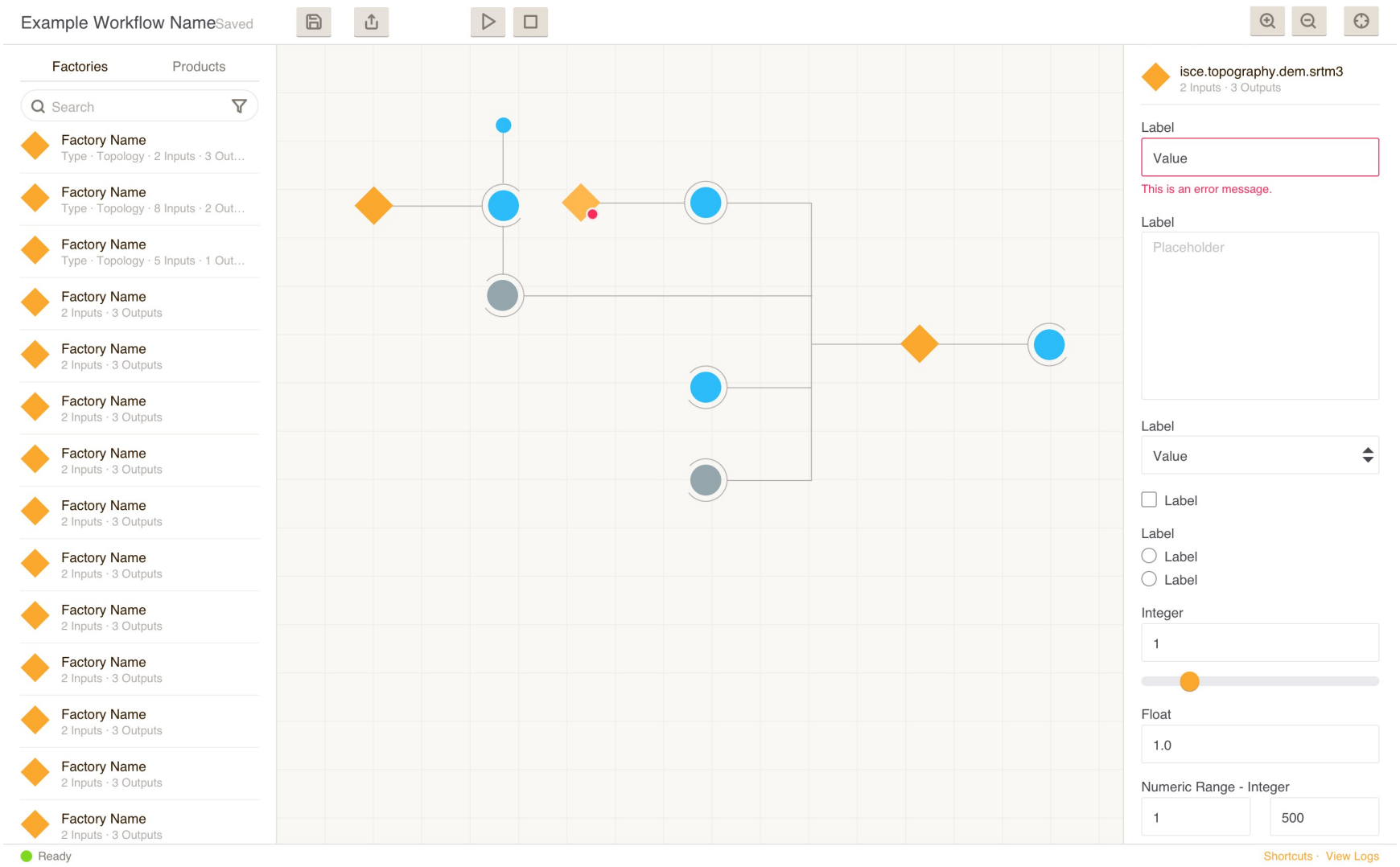
Client:

- SVG (and soon webgl) for the rendering of the graph
- ReactJS framework to build the user experience https://reactjs.org.

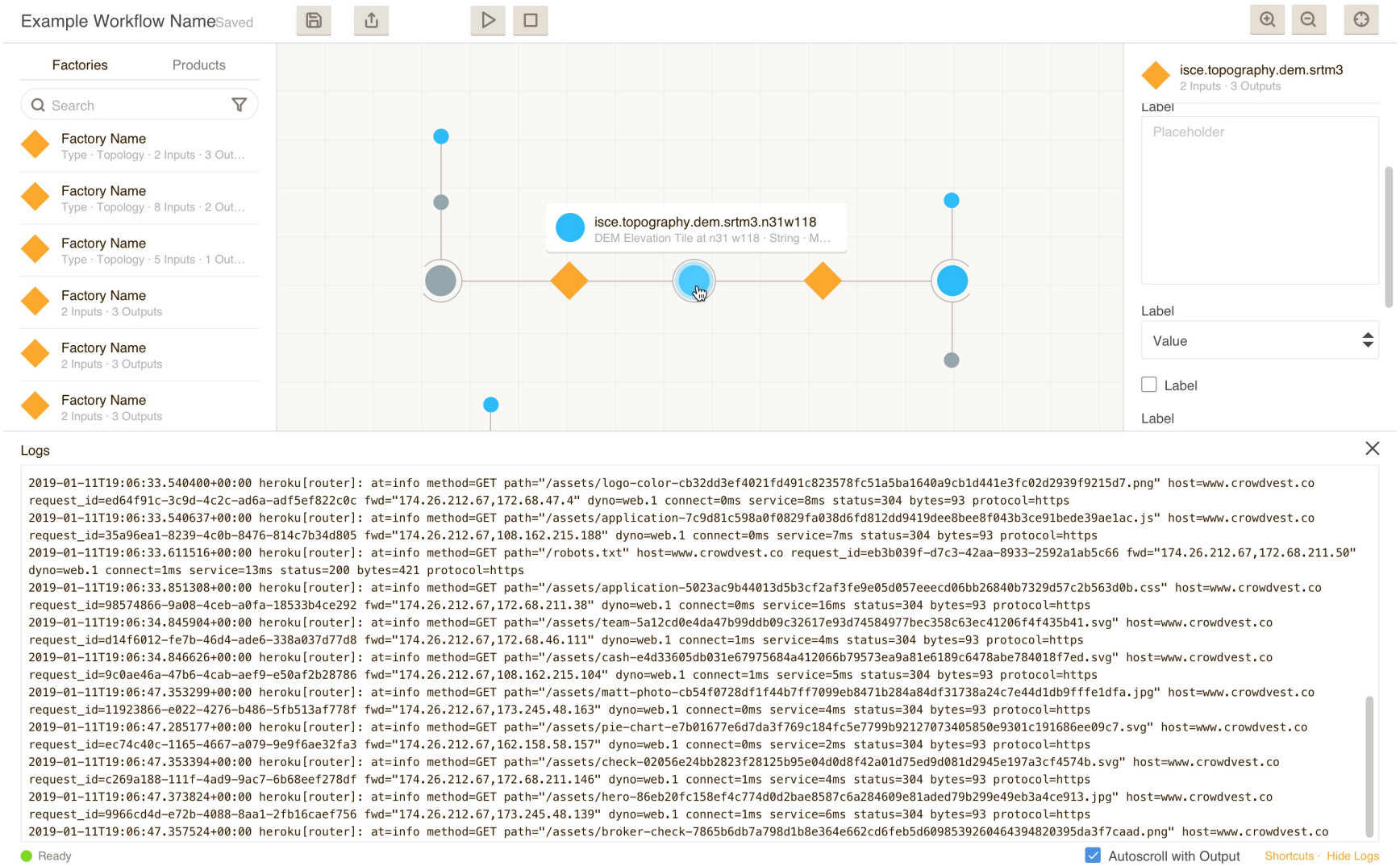
Client-Server interaction: a collection of data queries, mutations and subscriptions.

- GraphQL middleware (https://graphql.org).
- WebSockets.

Early design – editing factory metadata



Early design – flow execution



Summary

- ESTO Investments in SAR Processing are changing the way we interact with larger and larger volumes of data
- GPU-Accelerated generation of directly geocoded imagery enables previously prohibitive exploration of combinatoric interferometry
- GPU-Accelerated total workflow optimizations are enabling a degree of interactivity and experimentation on science problems
- A framework-based workflow visualizer, focused on products and interface simplicity, promises to make the interaction fun, iterative, and minable in the future