An Extensible Infrastructure for Processing Distributed Geospatial Data Streams*

Carlos Rueda, Michael Gertz, Bertram Ludäscher, and Bernd Hamann
Department of Computer Science
University of California at Davis
Davis CA 95616, U.S.A.
{carueda,gertz,ludaesch,bhamann}@ucdavis.edu

Abstract

Although the processing of data streams has been the focus of many research efforts in several areas, the case of remotely sensed streams in scientific contexts has received little attention. We present an extensible architecture to compose streaming image processing pipelines spanning multiple nodes on a network using a scientific workflow approach. This architecture includes (i) a mechanism for stream query dispatching so new streams can be dynamically generated from within individual processing nodes as a result of local or remote requests, and (ii) a mechanism for making the resulting streams externally available. As complete processing image pipelines can be cascaded across multiple interconnected nodes in a dynamic, scientist-driven way, the approach facilitates the reuse of data and the scalability of computations. We demonstrate the advantages of our infrastructure with a toolset of stream operators acting on remotely sensed data streams for real-time change detection.

1 Introduction

Advances in remote sensing research and technology have fostered a diverse spectrum of applications in the Earth sciences at multiple spatio-temporal scales. Besides global, long-term changing phenomena (*e.g.*, climate change), also important are the real-time settings where prompt response is of major concern, as is the case of environmental monitoring (*e.g.*, hurricane tracking and wildfire detection). Although increasingly larger sets of remotely sensed data are continuously streaming down to Earth, and computing infrastructures are improving, the necessity for specially tailored scientific data management frameworks has been recognized. Operational tasks like data handling and preprocessing, remote collaboration among researchers, and –in particular– data sharing, are an integral aspect of the scientific effort and should be considered in a systematic way.

*This work is in part supported by the NSF under awards IIS-0326517 (GEOSTREAMS) and EAR-0225673 (GEON).

Current satellite data delivery systems generally provide query capabilities for users to request specific products according to several criteria (*e.g.*, area of interest, time periods, spatial resolution). Complete datasets are generated and then delivered on demand. Less frequent is that users be able to access the raw streaming data generated by the satellite sensors. In spite of the current options to obtain data, scientists often find themselves replicating some of the required computations because of a lack of effective ways to discover existing datasets and easily integrate them into new computations (for example, cloud cover masks are a common input in many remote sensing processes).

We address the above shortcomings, specifically in the case of streaming geospatial image data, by proposing an extensible and scalable infrastructure for the processing of data streams. In our approach, nodes in a networked environment are set up to support the execution of streaming image pipelines with some components performing remote sensing analyses, and others enabling the remote access to the streams in the pipeline.

2 Background

Processing Remotely Sensed Data. At a high-level, image processing in remote sensing change detection typically includes the following steps: 1) data acquisition, 2) geometric rectification, 3) radiometric correction, 4) change detection, and 5) product generation. Primary inputs include reflectance information at various spectral wavelengths. Data must next undergo a number of preprocessing steps before it can be analyzed for change detection. Sub-setting (extraction of region of interest), masking, and mosaicking (combination of multiple images into a composite one) are routine pre-processing steps as datasets may originate from different sensors and/or be captured under different conditions. The images in the available datasets usually must be co-registered such that they are geometrically aligned, *i.e.*, pixels are located in the same reference system. Also differences in sensor characteristics (*e.g.*, radiometric precision, spectral response) must be rectified. From an abstract pers-

pective, the scientist chains these operations together in the form of direct acyclic graphs (DAGs), and then applies the necessary parameterization to accomplish a set of tasks.

Scientific Workflows. A scientific workflow is a collection of well defined activities and computations to attack a scientific problem. In this work we use KEPLER [8], a multi-institutional, open-source project aimed at offering scientists in diverse disciplines a system to design, execute, and deploy scientific workflows using Web- and Grid-based technologies. KEPLER is built on top of the PTOLEMY II system [4], which provides the infrastructure to support the execution of scientific workflows under a wide set of models of computation.

A scientific workflow is a network of *actors* whose mutual interaction is governed by a particular *model of computation* as implemented by a corresponding *director*. The overall structure is illustrated in Figure 1. Among the variety of models of computation provided by the underlying PTOLEMY II system, we use the process networks (PN) model as this naturally supports the execution of workflows on streaming data [7]. In the PN model, actors execute concurrently and communicate under a blocking-read, non-blocking-write scheme.

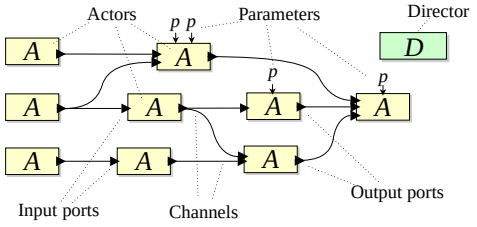

Figure 1. Main elements in a workflow.

The KEPLER/PTOLEMY II environment is an excellent vehicle for scientists to interact with our proposed infrastructure for a number of reasons. The toolset of streaming image processing operators developed in this work is integrated with the extensive list of KEPLER/PTOLEMY II actor packages that include operations for transparent access to Web services, Grid data transfers, remote job execution, and interaction with database management systems. Features like visual workflow design, ontology integration, and support for semantic types make the environment especially useful for domain scientists.

3 Stream Operators

To describe the data and the operations on remotely sensed image data, we use the algebra described by Gertz *et al* [2]. We define a *raster image* as a function from a *raster point set* $\mathbf{X}$ to a *value set* $\mathbb{F}$, where $\mathbf{X}$ is a regularly-spaced lattice in $\mathbb{R}^2$ or $\mathbb{R}^3$, and $\mathbb{F}$ is some algebraic system (a set together with a finite number of operations). Typical value sets in remotely sensed data are the integers, real numbers, as well as vectors of these types for *multi-banded* images. Being functions, common operations on functions are applicable to images as well. For example, a domain restriction on an image $\mathbf{a} : \mathbf{X} \rightarrow \mathbb{F}$ to a point set $\mathbf{R} \subseteq \mathbf{X}$, denoted $\mathbf{a}|_{\mathbf{R}}$, is defined as the function $\{(\mathbf{x}, \mathbf{a}(\mathbf{x})) \ : \ \mathbf{x} \in \mathbf{R}\}$. In our case, we often refer to $\mathbf{R}$ as a *region of interest* (ROI). Also, operations associated with the value set $\mathbb{F}$ *induce* corresponding operations on $\mathbb{F}$-valued images. For example, binary addition of real numbers induces the binary addition of real-valued images $\mathbf{a}$ and $\mathbf{b}$, formally expressed as $\mathbf{a} + \mathbf{b} \equiv \{(\mathbf{x}, \mathbf{a}(\mathbf{x}) + \mathbf{b}(\mathbf{x})) \ : \ \mathbf{x} \in \mathbf{X}\}$, where $\mathbf{a}, \mathbf{b} : \mathbf{X} \rightarrow \mathbb{R}$. Time is included in the model by associating a time stamp $\tau(\mathbf{a})$ to each image $\mathbf{a}$. A *raster image stream* (*stream*, for short) is a sequence of timestamped raster images $\alpha \equiv \langle \mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_t, \ldots \rangle$ where $\tau(\mathbf{a}_t) < \tau(\mathbf{a}_{t+1})$. Each image in a stream is assumed to be captured instantaneously. We use the notation $\alpha \equiv \langle \ldots, \mathbf{a}_t \rangle$ to refer to the *current image* $\mathbf{a}_t$ just produced by the data source generating the stream α. In this paper, we use the concept of current image as a convenient simplification to describe some of the operations on streams, particularly binary operations. An appropriate semantics actually involves considerations related to spatio-temporal granularities, synchronization, and handling of missing data. See [2] for details.

Operations on streams include those induced by operations on images. For example, spatial restriction of a stream $\alpha \equiv \langle \ldots, \mathbf{a} \rangle$ to a region of interest (point set) $\mathbf{R}$ is defined as $\alpha|_{\mathbf{R}} \equiv \langle \ldots, \mathbf{a}|_{\mathbf{R}} \rangle$; and the binary addition of two streams $\alpha \equiv \langle \ldots, \mathbf{a} \rangle$ and $\beta \equiv \langle \ldots, \mathbf{b} \rangle$ is the stream formed by adding their current images, $\alpha + \beta \equiv \langle \ldots, \mathbf{a} + \mathbf{b} \rangle$. Unary and binary operations on images (absolute value, addition, multiplication, etc.) are naturally extended to streams.

An important operation for temporal change detection is the *delay* operation. Given a stream $\alpha \equiv \langle \ldots, \mathbf{a}_t \rangle$, a *delayed* stream by δ samples is defined as $D^{\delta}(\alpha) \equiv \langle \ldots, \mathbf{a}_{t-\delta} \rangle$, *i.e.*, the current image in $D^{\delta}(\alpha)$ is the image that was current δ clocks ago in α. *Temporal restriction*, which selects the images in a stream whose timestamps are contained in a given set of time intervals, as well as *value transformation* and *spatial reprojection* (conversion from some coordinate system to another) are also typical operations on streams [2].

Figure 2 illustrates some of the operators in terms of workflow actors. The G-Reader actor can read a stream

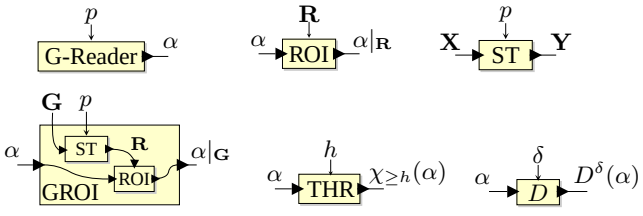

Figure 2. Operators in the toolset.

from the GEOSTREAMS server [3] or from other workflows as explained in Section 4. The ROI actor applies the spatial restriction of a stream α over a point set $\mathbf{X}$ to a given raster point set $\mathbf{R} \subseteq \mathbf{X}$. Regions of interest are very often specified in geographical coordinates, *e.g.*, latitude-longitude and Universal Transverse Mercator (UTM), rather than in specific sensor coordinates. Given an appropriate set of parameters p, the ST actor applies the corresponding spatial transformation to convert a point set $\mathbf{X}$ into a point set $\mathbf{Y}$. The GROI actor spatially restricts a stream α over a point set $\mathbf{X}$ to a given region of interest $\mathbf{G}$ that is given in some geographical coordinate system. The THR actor applies a thresholding operation to each image in a stream producing the corresponding stream of resulting images. Given a real-valued image on a point set $\mathbf{X}$, and a scalar $h \in \mathbb{R}$, the thresholding operation gets the binary image $\chi_{\geq h}(\mathbf{a}) \equiv \{(\mathbf{x}, 1) : \mathbf{a}(\mathbf{x}) \geq h\} \cup \{(\mathbf{x}, 0) : \mathbf{a}(\mathbf{x}) < h\}$. The last actor in Figure 2 performs the delay $D^{\delta}(\alpha)$ operation given a non-negative integer parameter δ.

4 Inter-Workflow Composition

We extend the composition mechanism described above to the inter-workflow level by means of inter-workflow connections. "Producer" workflows operate on the streams provided by primary data sources, *e.g.*, satellite receivers, and perform typical pre-processing operations including radiometric and geometric calibrations, and re-projection to commonly used spatial reference systems. Some of them may also provide storage related capabilities, *e.g.*, historical static datasets and delayed streams.

"Consumer" workflows ingest data from producer workflows (as well as from other external data sources), and typically perform tasks related to report generation, visualization, and even further processing to eventually become producers of new data streams.

Workflow Interconnection. A specially designed component, the *Net interface* (NI) actor, is used to indicate the streams in a workflow that are to be externally accessible. Actual access to the streams in a net-enabled workflow is accomplished by means of two special components that collaborate with the NI actor: a *query dispatch* (QD) actor and

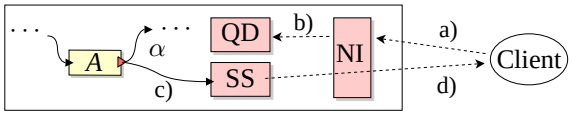

Figure 3. NI, QD, and SS collaboration.

a *stream server* (SS) actor. Figure 3 shows the collaboration between these components: a) a client requests stream α from the workflow; b) NI forwards request to QD, which returns output port for α; c) NI instantiates an SS actor; d) the SS sends requested stream to client. Only one SS instance is created for each stream; multiple clients interested in this stream are handled by the same SS instance.

Dispatch Extensions. As shown above, the NI dynamically instantiates SS actors to deliver data streams to clients. The QD has the same ability of dynamically adjusting the workflow structure while processing specific requests. Although a detailed description of a comprehensive query dispatching mechanism is beyond the scope of this paper, we illustrate the concept with the *region of interest* operation.

The query dispatch actor will create a new ROI actor instance for each new region of interest requested from the workflow. ROI actor instantiation is illustrated in Figure 4. To dispatch a new region of interest $\mathbf{R}$ on a given stream

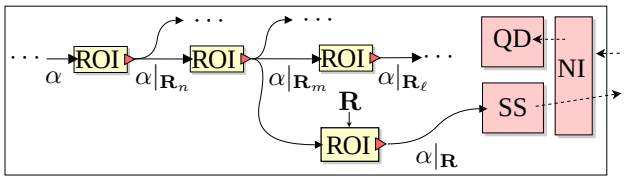

Figure 4. ROI actor instantiation.

α, QD checks the set of current streams to find the smallest $\mathbf{R}_m$ associated with α such that $\mathbf{R} \subseteq \mathbf{R}_m$. Initially, a source stream α (possibly unrestricted) would be chosen. Next, QD instantiates a new ROI actor with input $\alpha|_{\mathbf{R}_m}$ and parameter $\mathbf{R}$. NI gets the returned output port corresponding to $\alpha|_{\mathbf{R}}$, and instantiates a stream server actor to deliver the requested stream to the client.

Prototype. Actors for query dispatching, network interface, stream delivery, and several others for stream processing have been developed in Java using the PTOLEMY II libraries. We use real-time data streams from NOAA's GOES-10 satellite [9]. With raw data transmitted at 2.6 Mbps, the imager instrument in the GOES-10 satellite senses radiant and solar-reflected energy, generating data for five channels with different spatial resolutions and spectral characteristics. Visible channel data is generated at a rate of about 190Kbytes/second, taking about 26 minutes to scan the $22,600 \times 10,900$-pixel scene for the full field of view of the satellite. GOES data and derived products include surface temperature, cloud cover, and wildfire detection. Actual data from the GOES-10 satellite is provided in real time by the GEOSTREAMS system.

A complete image differencing and thresholding pipeline spanning two workflows is shown in Figure 5. The final stream obtained at the output of the THR actor in the consumer workflow (shown at the bottom of the figure) can be expressed as $\chi_{\geq h}(\alpha - D^1(\alpha))$ for a given threshold h. The producer workflow on the top of the figure makes available

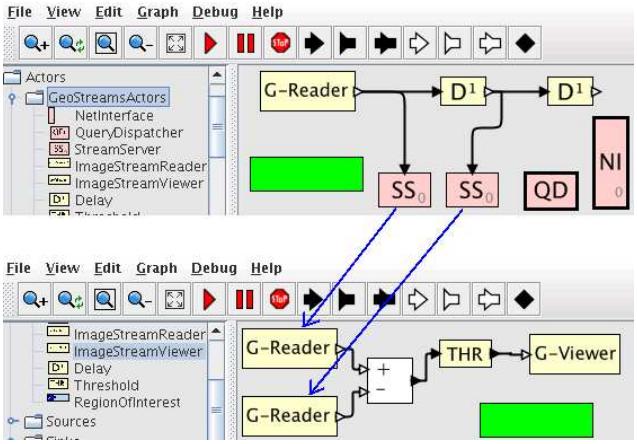

Figure 5. Producer and consumer workflows.

the stream α corresponding to the GOES-10 visible channel provided by the GEOSTREAMS server via a G-Reader actor, as well as the delayed streams $D^1(\alpha)$ and $D^2(\alpha)$. The consumer workflow uses two different G-Reader instances to import the streams α and $D^1(\alpha)$ from the producer workflow. It performs the image differencing and thresholding operations. It finally uses a visualizer actor, G-Viewer, also included in the toolset, for the real time visualization of the resulting thresholded stream. Some of the visualization functionality will also be demonstrated.

5 Related Work

Little work has been done for processing remotely sensed geospatial streaming data in a distributed, scientific workflow oriented fashion. Jaeger *et al* [5] propose to publish geospatial data and operations as individual Web services, which can then be composed using the KEPLER scientific workflow system. There are several fundamental differences to our approach: In [5], like in our approach, complex processing workflows execute via a process network director (PN) and exhibit *task parallelism*, *i.e.*, tasks on different branches of the workflow can run concurrently. However, in addition to task parallelism, we take full advantage of the process network model of computation [7] and provide real *pipeline parallelism*. This means that *all* instances of workflow actors (*i.e.*, components implementing stream operators) can execute simultaneously. In particular, consecutive actors on the same branch can work at the same time on different parts of the token stream. In contrast, most scientific workflow approaches, including [5], do not support pipeline parallelism. Often this is due to the fact that the Web services on which they are built do not support pipelining, or because the underlying model of computation does not support pipelined execution.

The main novelty of our approach is the ability to interconnect independently executing workflow instances, running on different sites. The different instances can publish and subscribe to each other's data streams, providing a simple yet flexible means to reuse live data streams. In this way, our approach is fundamentally different from conventional Web-service based approaches (such as [5]), and instead more similar to distributed environmental monitoring systems such as the Antelope Real-Time System [1].

6 Conclusions and Future Work

We have presented an approach for the extensible and scalable processing and analysis of remotely sensed streaming image data particularly focused on its usability by domain scientists. Our approach builds on KEPLER/PTOLEMY II, a robust, scientist-friendly problem-solving environment. Both data reuse and computational scalability are facilitated because complete processing image pipelines can be cascaded across multiple nodes. The mechanisms are readily extensible to allow incremental sophistication as it builds on the actor-oriented paradigm so they integrate naturally into the workflow framework. Our current efforts are aimed at enriching the toolset of change detection operators and tuning up their integration with the existing actor packages in KEPLER/PTOLEMY II.

References

[1] Antelope Environmental Data Collection Software. `http://www.brtt.com`.

[2] M. Gertz, Q. Hart, C. Rueda, S. Singhal, and J. Zhang. A Data and Query Model for Streaming Geospatial Image Data. In *Workshop on Foundations of Models and Languages for Data and Objects* (QLQP), 2006.

[3] Q. Hart and M. Gertz. Querying streaming geospatial image data: The GeoStreams Project. In *17th International Conference on Scientific and Statistical Database Management*. 2005.

[4] C. Hylands, E. Lee, J. Liu, X. Liu, S. Neuendorffer, Y. Xiong, Y. Zhao, and H. Zheng. Overview of the Ptolemy Project. Technical Report, University of California, Berkeley, 2003.

[5] E. Jaeger, I. Altintas, J. Zhang, B. Ludäscher, D. Pennington, and W. Michener. A Scientific Workflow Approach to Distributed Geospatial Data Processing using Web Services. In *17th Intl. Conference on Scientific and Statistical Database Management*, June 2005.

[6] J. R. Jensen. *Introductory Digital Image Processing: A Remote Sensing Perspective*. Prentice Hall, 2005.

[7] E. A. Lee and T. Parks. Dataflow Process Networks. *Proceedings of the IEEE*, 83(5):773–799, May 1995.

[8] B. Ludäscher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger-Frank, M. Jones, E. Lee, J. Tao, and Y. Zhao. *Concurrency and Computation: Practice & Experience, Special Issue on Scientific Workflows*, Scientific Workflow Management and the Kepler System. 2005.

[9] NOAA. National Oceanic and Atmospheric Administration GOES Geostationary Satellite. `http://www.goes.noaa.gov/`.