Earth Science Data Processing With Nextflow

Aron D. Bartle, Adam Thurston, J. Chris Currey
NASA Langley Research Center

Data-Driven Processing

This poster explores the use of Nextflow (www.nextflow.io) by the CLARREO Pathfinder (CPF) data management team.

Nextflow is an open-source tool for writing data-driven computational pipelines. It was initially developed for bioinformatics but is a good fit for other scientific domains (such as Earth Science) where…

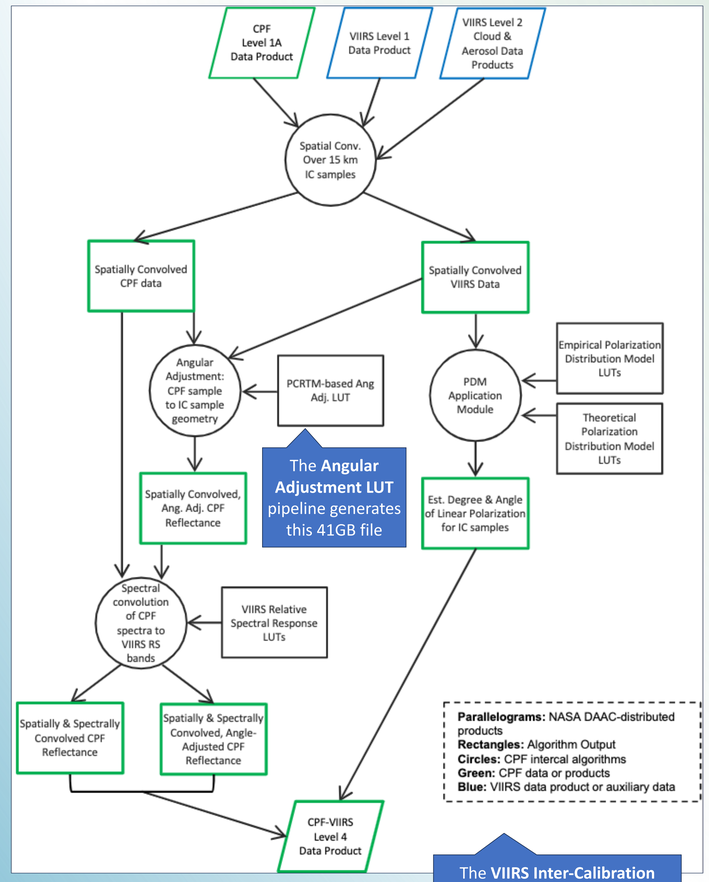
1. The problem to solve is a complex dataflow where multiple processing steps produce and consume data and must run in a specific order.
2. The processing steps require diverse languages, libraries, or tools.
3. There is a need to support multiple execution environments and scale to cloud resources.
4. There is a desire to follow principles of **open science** and build reproducible pipelines with low barriers to entry.

Nextflow is **not** an ideal choice where…

1. The solution requires very many short-lived processes (may cause inefficiencies when scheduling remote execution jobs).
2. There is a need for a workflow engine that natively supports standard workflow languages (CWL, WDL, etc.).

Additionally, Nextflow does not directly address use-cases related to scheduling or managing multiple pipeline executions.

CPF VIIRS Inter-Calibration Dataflow



The **Angular Adjustment LUT** pipeline generates this 41GB file

The **VIIRS Inter-Calibration** pipeline solves this dataflow

About Nextflow

Nextflow pipelines are programs written in a domain-specific scripting language (an extension to Groovy). Pipelines define:

- *Processes* – individual processing steps, written in any scripting language (Bash, Perl, Python, R, etc.).
- *Workflows* – compositions of processes and dataflow logic that achieve some goal.

All dataflow logic is accomplished via *channels* and channel *operators*:

- *Channels* – queues that connect to/from processes and emit *items* (collections of files and/or values).
- *Operators* – methods that create new channels by transforming items from existing channel(s).
  For example: "**z = x.mix(y)**" – channel z emits items from channels x and y.

All processing is *reactive*:

- processes run whenever a data item is available in their input channel(s).

Nextflow and Open Science

1) Install

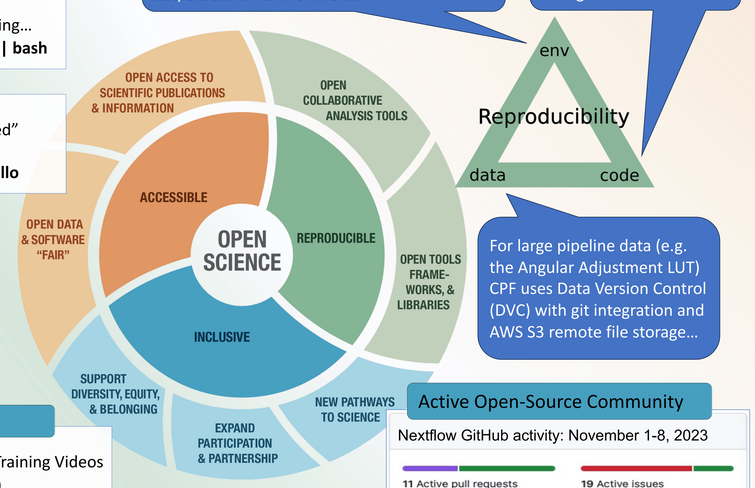
Nextflow runtime is self-installing…
curl -s https://get.nextflow.io | bash

2) Run

Shared pipelines may be "pulled" from git repositories…
./nextflow run nextflow-io/hello

Process scripts run in the context of a software container (e.g. Docker) to ensure a consistent test/execution environment…

Pipelines are stored as git repositories, encouraging strong version control…



For large pipeline data (e.g. the Angular Adjustment LUT) CPF uses Data Version Control (DVC) with git integration and AWS S3 remote file storage…

Open Resources

- Reference Documentation & Training Videos
- Slack Channel (ask questions!)
- Implementation Patterns
- **nf-core** – curated shared pipelines
- **GitHub** – open development platform

Active Open-Source Community

Nextflow GitHub activity: November 1-8, 2023

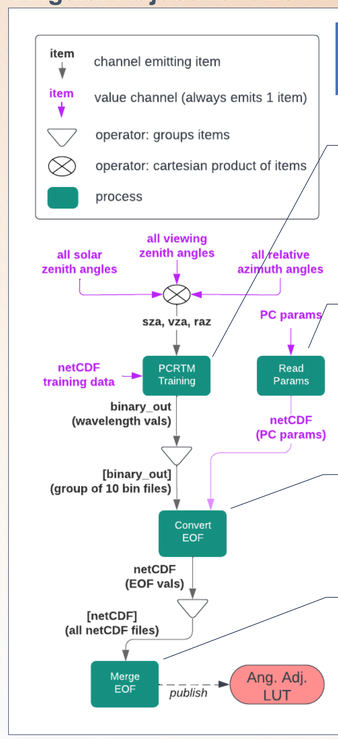
11 Active pull requests	19 Active issues		
4 Merged pull requests	7 Open pull requests	14 Closed issues	5 New issues

NASA "Open Science" Image From: https://www.earthdata.nasa.gov/technology/open-science

CLARREO Pathfinder Pipelines

Pipeline	File Produced	CPU-hours	Time to Completion (AWS Batch)	Processes	Total Process Invocations
CERES Inter-Calibration	Level 4	0.2	19 min	8	9 - 10
VIIRS Inter-Calibration	Level 4	0.6	27 min	15	44 - 68
CLARREO Simulator	Level 1A	~800	~60 min	14	~44
Angular Adjustment LUT	Look-up Table	~700,000	~10,000 min	4	20,515

Angular Adjustment LUT Pipeline Design



This pipeline generates the Angular Adjustment Look-Up Table (LUT) used in CERES/VIIRS Inter-Calibration pipelines.

PCRTM Training runs a Fortran radiative transfer simulation for each combination of viewing angles, using a netCDF file containing training observations.

18,648 invocations = 24 sza * 21 vza * 37 raz

Read Params reads Principal Component (PC) analysis parameters into a netCDF file.

1 invocation

Convert EOF runs a Python script to convert PCRTM binary radiance values (in groups of 10 files) into the Empirical Orthogonal Function (EOF) domain and reformat as intermediate netCDF files.

1,865 invocations = ⌈18648/10⌉

Merge EOF runs a Python script to combine all intermediate netCDF files into a single LUT.

1 invocation

Nextflow Attributes

Scalability

- Scales at the level of a process invocation.
- Supports batch schedulers (SGE, Slurm, PGE), cloud batch services (AWS, Azure, Google), distributed clusters (Kubernetes), and others.
- NF runtime schedules remote jobs and ensures channel data (items) go where they are required.

Portability

- NF runtime requires only Java 11 or later.
- Integrates with package managers (e.g. Conda) and software containers (e.g. Docker).
- Config *profiles* isolate non-portable settings.
  - Same pipeline runs in many execution environments. CPF pipelines are configured for laptops and AWS Batch.

Usability

- Processes don't require APIs/interfaces.
- Every process invocation has a unique work folder with full context – aids debugging.
- HTML reports with process statistics: CPU and memory usage, job duration, and I/O performance.
- Logging, trace, execution graph visualizations.

Reliability

- Pipelines specify how processes respond to errors.
- Ability to resume long-running failed or cancelled pipelines.
- Large and active user base signals robustness. *The CPF team reported issue #2169 on 6/15/2021… fixed in release dated 8/6/2021.*

Conclusions

Lessons Learned

- Solving dataflow logic with channel operators is challenging at first.
  - Learn the implementation patterns (GitHub *nextflow-io/patterns*)
- Implement complex algorithms as external libraries (e.g. Python modules), add to Docker image, import & use in process script.
- Run pipelines in AWS Fargate containers for serverless deployments.
- Tailor AWS Batch configuration to unique process requirements:
  - "PCRTM Training" jobs use 48 vCPU instances with spot pricing.
- Use JSON-based data product metadata to write processes that support heterogeneous Earth Science data products.

Future Work

Develop shared resources for pipeline developers using netCDF files and Earth Science data products:

- Modules – merging netCDF files, normalizing netCDF variables.
- Patterns – parallelization over netCDF data, JSON data product metadata.