ALGORITHM PROCESSING WITH SEED AND SCALE

VALIDATE AND ENABLE YOUR DATA

INTRODUCTION

Seed and Scale are part of the National Geospatial-Intelligence Agency's (NGA) suite of tools that assist in the full lifecycle of algorithm development. This suite provides data scientists and algorithm developers with the ability to complete algorithm proof of concepts, test and refine algorithms over time, validate new algorithms against massive volumes of historic data and integrate final algorithms into a workflow. It allows decision makers to validate data flows and look for consistent data volumes. Applied Information Sciences (AIS) is the recognized expert in the development, maintenance and upgrades to both Seed and Scale. This white paper outlines our current knowledge of Scale and Seed, the roadmap for near-future updates, lessons learned, and our proposed solution customized to your needs.

SEED

Seed is a standard to aid in the discovery and consumption of a discrete unit of work contained within a Docker image. The Seed standard clearly defines the requirements and interface of an algorithm, ensuring that local testing is consistent with operational environments. We provide a Command-Line Interface (CLI) that is useful to ensure Seed specification compliance and facilitates testing of the algorithm according to the defined interface.

A Seed compliant algorithm:

- Can be easily used on any supported platform (Linux / Mac / Windows)
- Includes an embedded self-describing and intuitive manifest
- Can be efficiently integrated into Scale
- Allows for discovery from public Docker images hosted within either Docker Hub, Docker Registry or Container Yard.

SCALE

Scale enables near real-time automated processing of large data sets and allows algorithms to be run in a distributed processing cluster. It provides a framework and User Interface (UI) to integrate, manage and monitor algorithms, data and computer resources. It makes cloud development, testing, and operational monitoring of geospatial algorithms seamless and efficient and enables a true DevOps environment for algorithm development and cloud transition.

Scale UI provides an interactive dashboard that enables users to monitor their algorithms and resource utilization and includes powerful visualization components that allow users to chain together multiple algorithms (called a Recipe).

BENEFITS OF SCALE

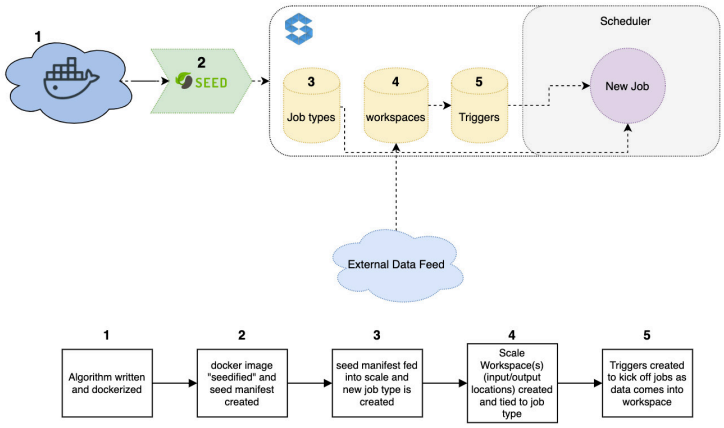
With Scale you will benefit from:

- The ability to execute large-scale data ingestion utilizing a combination of C2S services, including both block (EFS/EBS) and object storage
- Automatically scheduling new data for processing against assigned algorithm workflows
- Ability to publish results correlated to job and workflow
- Rapid validation of new algorithms against massive volumes of historic data
- A powerful platform for managing cluster resources to make processing resilient and cost-effective, both on-premise and in the cloud (C2S)
- Support for multiple sensor feeds allowing for independent metrics on data volume, file count and derived products

THE CURRENT ECOSYSTEM

When designing Seed, our goal was to resolve algorithm integration problems as close to the developer as possible. Developers can leverage Seed CLI to locally mirror the runtime environment found within Scale. This will ensure the ability to properly build a final artifact that can be successfully orchestrated within Scale. All jobs that run within Scale are fundamentally Docker images. Seed merely provides a manifest (defined as a Docker Label) that extends the Docker image to advertise the algorithm interface.

Lesson Learned: While onboarding several external teams we found that the addition of Seed greatly improved the Scale onboarding process. It allows external teams to focus on their individual processing algorithms without the complexities of managing a distributed cluster environment. Once these algorithms are validated locally and published, we work in tandem with the developer to support defining workflows and configuring operational data feeds to Seed compliant jobs. We feel that it is important to work closely with new teams to maximize the overall user experience of the system.



Scale allows for complex workflows to be defined using the “recipe” concept. These recipes make it possible to perform complex branching and conditional operations using the Scale Recipe Domain Specific Language (DSL). With these workflows, jobs can be launched or omitted based on upstream data processing. Throughout the workflow, source data can be enriched with extracted metadata.

THE USER EXPERIENCE

Scale provides an interactive dashboard that enables users to monitor cluster status, jobs and resource utilization. The Scale web application serves both administrators, data scientists and decision makers with job history, data lineage, source and product data access and system health. Users can quickly analyze the status of their recipes and jobs through tables and metric reporting that allow for data to be filtered by date and time. The dashboard experience can also be customized to “favorite” jobs that mean the most to each viewer.

Seed compliant algorithms, whether written in Java, Python, IDL, MATLAB, C/C++ and more, equate to rapid integration with Scale. The UI includes powerful visualization components that chain together algorithms via recipes and allow developers to run their algorithms concurrently or synchronously on files. The Recipe Builder allows analysts to customize their algorithm workflow to optimize the experience for you and your team. Recent UI improvements include compatibility with Seed and improved authentication capabilities.

Looking Forward: AIS continues to strive for a personal user experience. In the near future we plan to customize the interface to allow for different team roles (system administration and algorithm development), improved navigation structure and additional functionality within the recipe editor including a recipe wizard for first-time or infrequent users.

MAINTENANCE

AIS has been actively involved in the development of Scale since the project’s inception. As a result, we have an unmatched knowledge of the daily tasks that are required to provide consistent, optimal performance from the system. Most maintenance issues with Scale are tied to the type of jobs that are being processed by the system. By design, Scale will attempt to resolve systemic problems without user intervention. For example, if a single node is performing erratically, Scale will flag it as degraded and cease scheduling work to that node. Any jobs that failed on that node will be moved to healthy nodes to complete. **With our baseline production Scale cluster, the recurring maintenance load is 4-8 hours from 1 FTE per week.**

Lesson Learned: Over time we have devised exhaustive checklist procedures that help us detect problems early and avoid any truly catastrophic disaster recovery events.

SYSTEM THROUGHPUT

Our baseline production Scale cluster is deployed in C2S and is responsible for near-real time processing of all collections from multiple intelligence community sensor platforms. These feeds vary in size based on various events, but presently **we are responsible for between 10 and 100TiBs daily in data volume**. This does not include the derived products created as output from these source datasets.

There are many individual algorithmic workflows performing processing based on the collection type detected. **We average 30,000 jobs launched per day and regularly burst up to 10,000 jobs per hour**. Based on server analytics, **we serve approximately 1500 unique users on a weekly basis**.

SOFTWARE LICENSING

There is no hard requirement from the current system for licensed software. Scale was built at NGA to ensure frictionless deployments of any size. This motivated the use of software and tools free of license encumbrance. Scale is licensed under the Apache 2.0 license and minimally requires D2IQ's DC/OS, Docker CE, PostgreSQL / PostGIS, Elasticsearch, RabbitMQ and HashiCorp Vault. Each of these can be acquired as free open-source software or as enterprise offerings including support. There are certainly advantages in using the enterprise offerings, but our approach has been to use cloud managed offerings where available.

THE FUTURE OF SCALE AND SEED

Scale was built to optimize the utilization of on-premise hardware with file-based feeds of data. By leveraging Docker and Apache Mesos, newly available at that time, Scale was able to improve utilization by an order of magnitude. As C2S became available, Scale was augmented to offer native support for AWS storage and messaging services. As Scale has moved into the cloud for the majority of installations, the Apache Mesos abstraction for hybrid deployments has become less advantageous. Our goal for future releases of Scale is separate the workflow engine and processing queue from the platform orchestration. This will allow Scale to continue to operate on Apache Mesos, as well as Kubernetes or natively on cloud provider infrastructure.

The AIS team has been tasked by our NGA client to make the system cloud native, first focusing on one platform. As part of that effort we plan to evaluate the system architecture to better streamline the design, incorporating the most desirable features that have been added to Scale over the years. We want to further improve the user experience to center the interface, tasks, and flow in the UI around the user's goals.

SOLUTION APPROACHES

ALGORITHM OPTIMIZATION

AIS will work with you to complete an algorithm optimization trade study which will help you configure the system to best meet your needs. As part of the study, we would consider the following high-level guidelines:

- **Workflow Decomposition:** Breaking down larger processing chains into discrete units of work (algorithms) has numerous advantages that will yield gains throughout the development lifecycle. While this is not always possible when algorithms are delivered in a complete state, it is something that should be a top priority. Not only will problems be discovered sooner, it will allow component algorithms to be built and tested independently. This will help to minimize any single person or team being a bottleneck. ***The greatest gains may be had when extracting general purpose, reusable algorithms that can be applied to multiple data types***, such as a simple algorithm for generating a tile pyramid within a GeoTIFF. This will ensure that commonly used algorithms are both optimized and thoroughly proven by reuse.

- **Logging:** Very few things are more frustrating than having a problem within a piece of software that fails without any indication of what went wrong. When running algorithms inside of containers on a distributed system, it becomes difficult and sometimes impossible to perform interactive debugging. This makes good logging and error handling an absolute necessity. All that is necessary when using the Scale system is to write log messages to the console and they will be captured for live review.

- **Errors:** Specific error conditions indicated as unique process exit codes may also be used to give clear feedback when a known error has been encountered. These are commonly used for data errors where the data arrived but contained an error condition that could not be automatically addressed within the algorithm.

- **Runtime Configuration:** Most algorithms have configuration values that are desirable to change at runtime. This may be anything from confidence values for feature detection to band number within a multi-band image. ***Ensuring that your algorithm can consume its configuration as either environment variables or command-line arguments is our recommended best practice***. This ensures that your algorithm image is configurable without creation of a new Docker image for common changes.

- **Contextual Sizing:** As outlined in the first section, proper decomposition of a workflow is paramount to enabling reuse of algorithms and optimal team effectiveness. Understanding the type of machines that are being targeted to execute your workloads is important as well. When using many small VMs as opposed to fewer large VMs, you can expect considerably higher overhead for both Docker image and data retrieval. This can be further aggravated when using machines that have limited throughput of their network interface. It is generally advantageous to use machines large enough to accommodate 10 gigabit network interfaces. As a result, it is often only worth optimizing for input file size and runtime rather than for memory or CPU requirements.

- **Image Optimization:** It is very common to see use of Anaconda and other rather large libraries associated with data scientist and machine learning workloads. This makes sense for development on Windows or other environments that make compilation of complex scientific and math libraries challenging, but this paradigm should not be carried over to final Docker image artifacts. In order to avoid undue strain on Docker registry and image cache resources, images should be kept under 500MB as a rule of thumb. Docker images should be based on minimal operating systems (Alpine / Busybox) and layer optimization techniques (manual and automated) should be applied.

KUBERNETES EVALUATION

We will assist you in evaluating Kubernetes as part of the solution. If transitioning to Kubernetes, we would evaluate the current Scale architecture to optimize performance using platform concepts (Pods, Services, etc.). We would create a roadmap for abstracting core scheduling concepts used with DC/OS to aid in the transition to Kubernetes.

For our NGA client's work in making Scale cloud native we evaluated Kubernetes and several cloud platforms. We will leverage this knowledge to help you make the best decision for your environment and team.

INTEGRATION OF SCALE

AIS can work with your team to develop and implement a plan to deploy Scale. We will work with you to understand any constraints of your environment, determine how they impact Scale, and incorporate those into the plan. A common pattern is to install and configure independent Scale environments in order to execute baseline workflows while supporting a research and development environment for algorithm developers. This will allow your algorithm developers to familiarize themselves early with the system and begin testing and validation of their algorithms while we build an optimized AWS, Azure, or Kubernetes-based solution.

FEATURE REQUESTS

Finally, as part of our solution we propose that we include a user experience expert and a technical lead to work directly with your team and the algorithm developers. We will incorporate feature requests based on your needs and ensure the Scale baseline includes these updates. By providing a team of experts to you that also work directly with our NGA client, you can be confident that we can coordinate and prioritize your needs with accompanying Scale releases.

CONCLUSION

NGA's suite of Scale and Seed will enable on-demand, near real-time automated processing of large datasets and will allow you to validate, design and test algorithms in both a local and distributed environment. As the recognized experts with Scale and Seed, Applied Information Sciences is the right team to assist you with this project. We pride ourselves in providing professional, passionate technical experts that will work alongside you to develop the best solution for your needs. We look forward to working with you to migrate your algorithm workflows and provide an optimal processing experience.

www.ais.com

11440 Commerce Park Drive, Suite 600, Reston, VA 20191
Phone (703) 860-7800
sales@ais.com